

Réalisation d'un système d'information géographique de gestion d'infrastructures
routières du Québec et de calcul d'itinéraire en cas d'effondrement de ponts

par

Mahamat Imrane Khalid Abdelkerim

Mémoire présenté au Département d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, Avril 2019

Le 26 Avril 2019

Le Jury a accepté le mémoire de Monsieur Mahamat Imrane Khalid Abdelkerim dans sa version finale

Membres du jury

Professeur Gabriel Girard
Directeur de recherche
Département d'Informatique

Professeur Goze Bertin Benie
Codirecteur de recherche
Département de Géomatique appliquée

Professeur Luc Lavoie
Membre interne
Département d'Informatique

Professeur Bessam Abdulrazak
Président-rapporteur
Département d'Informatique

Sommaire

La bonne circulation des personnes et des biens est devenue un besoin essentiel de notre société. Le réseau routier est un élément important pour la croissance économique ; il constitue un indicateur primordial de la santé économique d'une région, car il sert tant au déplacement des personnes et des biens, qu'à la prestation des services tels que la collecte des déchets dans les centres urbains.

Or, il existe plusieurs éléments qui contribuent à la dégradation des infrastructures routières et nuisent à la bonne circulation. Parmi ces éléments on peut citer l'augmentation de la mobilité, les catastrophes naturelles et la vétusté de ces infrastructures routières. Il survient également des incidents qui perturbent la circulation, on peut citer le risque d'effondrement des ponts ainsi que les travaux sur les routes, qui entraînent des fermetures nécessitant une prise en charge afin d'assurer le service offert par le réseau routier malgré les difficultés présentes.

Plusieurs études ont montré que la croissance économique est liée à la qualité des infrastructures routières, donc il est important d'avoir des infrastructures routières bien aménagées et entretenues.

Parmi les solutions envisageables pour mieux gérer et entretenir les infrastructures routières, les Systèmes d'Information Géographique (SIG) semblent être un choix de premier ordre. C'est ainsi que dans ce travail nous avons construit un système d'information géographique Web de gestion des infrastructures routières et de gestion du trafic routier en cas d'incident.

Remerciements

Mes remerciements vont tout d'abord à Dieu qui m'a donné la santé et la force nécessaire pour arriver à terme de ce travail.

Je tiens également à remercier mes directeurs Gabriel Girard et Goze Bertin Benie pour toute leur assistance et accompagnement notamment les conseils, les orientations et les suivis permanents.

Mes remerciements vont également au Professeur Patrick Paultre directeur du CRGP qui a financé ce projet.

Je remercie également le jury qui a accepté d'examiner ce travail.

Je leurs exprime à tous ma profonde gratitude.

Je remercie aussi les professeurs Yves Voirin et Atilio François pour leur assistance. Sans oublier tous ceux qui ont contribué de près ou de loin sous toute forme à l'accomplissement de ce travail.

Finalement je remercie mes chers parents qui, grâce à eux que je suis ce que je suis ainsi que toute ma famille et toutes les personnes que j'ai côtoyées durant ma maîtrise.

Table des matières

Sommaire	iii
Remerciements	iv
Liste des tableaux.....	viii
Liste des figures.....	ix
Introduction.....	1
Mise en contexte et problématique	1
Objectifs de ce travail	3
Hypothèses	3
Plan du mémoire	4
Chapitre 1 Notions	5
1.1 Information géographique	5
1.2 La cartographie	6
1.2.1 La carte.....	6
1.2.2 Système de Coordonnées de Référence (SCR).....	6
1.3 Systèmes d'information géographiques (SIG)	7
1.3.1 Formats des données	8
1.3.1.1 Format raster	8
1.3.1.2 Format vecteur	9
a) Description	9
b) Format Shapefile	10
1.3.2 Les couches.....	11
1.3.3 Normes Open Geospatial Consortium (OGC)	11
1.3.3.1 Services.....	12
a) Web Map Service (WMS) [43]	12
b) Web Feature Service (WFS) [63]	13
1.3.4 Client Web (OpenLayers).....	14
1.4 Les graphes	15
1.4.1 Recherche dans un graphe.....	15
1.4.1.1 Parcours en largeur.....	16
1.4.1.2 Parcours en profondeur.....	16
Chapitre 2 Etat de l'art	18

2.1 Gestion des infrastructures routières	18
2.1.1 Les données	18
2.1.1.1 Données pour l'identification des infrastructures routières.....	18
2.1.1.2 Données d'inspection d'un pont pour la maintenance	21
2.1.1.3 Indicateurs de détérioration	22
2.1.1.4 Cause des détériorations des infrastructures routières.....	25
2.1.1.5 Intérêt des SIG.....	25
2.1.2 Les technologies et l'architecture.....	26
2.1.2.1 Caractéristiques des SIG Web	26
2.1.2.2 Composants de base de l'architecture SIG Web	28
2.1.3 Etude de l'existant	29
2.1.3.1 Travaux sur les SIG Web	29
2.1.3.2 Exemples des projets SIG pour la gestion des infrastructures routières	30
2.2 Optimisation des itinéraires et cartographie	31
2.2.1 SIG et Réseau routier.....	31
2.2.2 Problème de recherche du meilleur chemin.....	33
2.2.3 Algorithme de routage	34
2.2.4 Astuces d'accélération.....	35
2.2.5 Exemple d'études sur le calcul d'itinéraires	38
2.2.6 Applications cartographiques de calcul d'itinéraire	39
Chapitre 3 Conception et réalisation	45
3.1 Méthodologie.....	46
3.2 Analyse des besoins	46
3.3 Analyse et conception du système	47
3.4 Analyse des informations.....	48
3.4.1 Modèle conceptuel.....	48
3.4.2 Modèle logique.....	52
3.4.3 Intégration des données	53
3.5 Implémentation.....	54
3.5.1 Préparation des données	55
3.5.2 PostGIS/PostgreSQL.....	55
3.6 Serveur cartographique.....	60
3.6.1 Concept de GeoServer.....	60
3.6.2 Implémentation.....	61

3.6.3 Code Serveur	64
3.7 Client	65
3.7.1 Implémentation	65
3.8 Difficultés rencontrées	72
Chapitre 4 Calcul d'itinéraire.....	74
4.1 Module de calcul d'itinéraire	74
4.2 Principe de fonctionnement	75
4.3 Solution de calcul d'itinéraire	76
4.4 Implémentation	77
4.1.1 Création d'une fonction de calcul d'itinéraire	77
4.1.2 Création d'une vue SQL	78
4.1.3 Paramétrage de la vue	79
Chapitre 5 Résultats	81
5.1 Démarrage et accueil	81
5.2 Tests de consultation	82
5.3 Tests de calcul d'itinéraires	85
5.4 Analyse des résultats	90
Chapitre 6 Conclusion	92
Annexe A Présentation des données	94
A.1 Tableau 9 - Réseau routier (Source de données, Cartothèque de l'UDS et ministère des Transports, Division de la géomatique, 2015).....	94
A.2 Tableau 10 - Réseau routier (Source de données, OSM)	97
A.3 Tableau 11 - Ponts (Source de données, initiateurs du projet).....	97
A.4 Tableau 12 - Ponts (Source de données, Site web du Ministère des Transports, Division de la géomatique, 2015)	98
Bibliographie	100

Liste des tableaux

Tableau 1 - Données pour la maintenance d'un pont, d'un tunnel et d'une chaussée.....	20
Tableau 2 - Interprétations des valeurs de prédiction d'une chaussée	23
Tableau 3 - Interprétations des valeurs de prédiction d'un pont	23
Tableau 4 - Interprétations des valeurs de prédiction d'un tunnel	24
Tableau 5 - Réseau routier (Source de données, Cartothèque de l'UDES et ministère des Transports, Division de la géomatique, 2015).....	59
Tableau 6 - Réseau routier (Source de données, Open Street Map).....	59
Tableau 7 - Ponts (Source de données, initiateurs du projet).....	59
Tableau 8 - Ponts (Source de données, Site web du Ministère des Transports, Division de la géomatique, 2015).....	60
A.1 Tableau 9 - Réseau routier (Source de données, Cartothèque de l'UDS et ministère des Transports, Division de la géomatique, 2015).....	94
A.2 Tableau 10 - Réseau routier (Source de données, OSM)	97
A.3 Tableau 11 - Ponts (Source de données, initiateurs du projet).....	97
A.4 Tableau 12 - Ponts (Source de données, Site web du Ministère des Transports, Division de la géomatique, 2015)	98

Liste des figures

Figure 1 - Image d'un format shapefile muni des fichiers complémentaires, capturée dans Windows	10
Figure 2 - Superposition des couches.....	11
Figure 3 - Capture d'une requête WMS	13
Figure 4 - Capture d'une requête WFS.....	14
Figure 5 - Parcours en largeur (figure tirée du Cours IA A. Cornué jols).....	16
Figure 6 - Parcours en profondeur (figure tirée du Cours IA A. Cornué jols)	17
Figure 7 - Exemple de calcul d'itinéraire de Google Maps [30]	40
Figure 8 – Calcul d'itinéraire par Mappy (source : https://fr.mappy.com)	42
Figure 9 – Calcul d'itinéraire par Bing Maps [38]	43
Figure 10 – Ville de Québec.....	45
Figure 11 - Architecture du système	48
Figure 12 - Modèle conceptuel de données.....	50
Figure 13 - Modèle conceptuel de données (suite 1).....	51
Figure 14 - Modèle conceptuel de données (suite 2).....	51
Figure 15 – Modèle conceptuel de données (suite 3)	52
Figure 16 – Modèle logique de données.....	53
Figure 17 - Architecture de PostGIS	56
Figure 18 - Interface de pgAdmin	57
Figure 19 - Résumé des outils utilisés.....	58
Figure 20 - Interface de GeoServer	62
Figure 21 - Étapes de création d'une couche.....	63
Figure 22 – Vue de nos ressources dans GeoServer.....	64
Figure 23 - Code de recherche des points	65
Figure 24 – Création d'une couche	66
Figure 25 – Requête via le protocole HTTP.....	66
Figure 26 – Création d'une couche	67
Figure 27 – Création d'une carte.....	67
Figure 28 – Affichage des données en couches.....	68
Figure 29 – Contrôle de l'interaction utilisateur	69
Figure 30 – Affichage de résultat à l'utilisateur.....	69
Figure 31 - Création d'une couche	70
Figure 32 – Création d'une couche (suite)	71
Figure 33 – Création d'une couche (suite)	72
Figure 34 - Structure de pgRouting.....	75
Figure 35 - Données de calcul d'itinéraire	76
Figure 36 – Code d'exécution de A*.....	78
Figure 37 - Création d'une vue SQL	79
Figure 38 - Paramètres de la vue	80
Figure 39 - Interface du système	82
Figure 40 - Consultation d'un tronçon de route	83
Figure 41 - Consultation d'un tronçon de route (suite)	83
Figure 42 - Consultation de caractéristiques d'un pont vert.....	84
Figure 43 - Consultation de la courbe de fragilité d'un pont vert	84

Figure 44 - Consultation de caractéristiques d'un pont marron	85
Figure 45 - Consultation de caractéristiques d'un pont marron (suite)	85
Figure 46 - Calcul d'itinéraire	86
Figure 47 - Calcul d'itinéraire avec blocage d'un tronçon.....	86
Figure 48 - Calcul d'itinéraire avec blocage d'un autre tronçon.....	87
Figure 49 - Calcul d'itinéraire avec blocage de deux tronçons	87
Figure 50 - Calcul d'itinéraire avec blocage de deux autres tronçons.....	88
Figure 51 - Calcul d'itinéraire avec blocage de trois tronçons.....	88
Figure 52 - Calcul d'itinéraire avec blocage de trois autres tronçons	89
Figure 53 - Calcul d'itinéraire avec blocage de quatre tronçons.....	89
Figure 54 - Calcul d'itinéraire avec blocage de quatre autres tronçons	90

Introduction

Mise en contexte et problématique

De nos jours, la bonne circulation des personnes et des biens est devenue un besoin essentiel de notre société, tant sur le plan social que professionnel. En effet, selon des études [17] [20], le réseau routier est un élément important pour la croissance économique ; il constitue un indicateur primordial de la santé économique d'une région, car il sert tant au déplacement des personnes et des biens, qu'à la prestation des services tels que la collecte des déchets dans les centres urbains.

Or, il existe plusieurs éléments qui contribuent à la dégradation des infrastructures routières et nuisent à la bonne circulation. Parmi ces éléments on peut citer la croissance démographique, l'amélioration du niveau de vie des citoyens et le développement du secteur industriel qui, combinés, augmentent la mobilité et impliquent la présence d'un nombre important de véhicules sur les routes. Selon le *Journal de Montréal*, paru le 16 janvier 2018, il y a eu une augmentation de 15% du nombre d'automobiles dans la région de Montréal sur la période allant de 2007 à 2016, soit 8 168 750 de plus. D'autres éléments pouvant contribuer à la dégradation des infrastructures sont les catastrophes naturelles, les produits de déneigement et le vieillissement qui se manifestent par l'apparition de nids de poules, de fissuration, d'affaissement et même d'effondrement de ponts.

Ainsi, tous ces facteurs peuvent provoquer des problèmes de circulation importants, qui à leur tour induisent des conséquences psychologiques, économiques, environnementales et sécuritaires. Ces conséquences se manifestent par des pertes de temps dues aux congestions, une consommation supplémentaire de carburant, une augmentation de l'émission de gaz à effet de serre, un nombre plus élevé d'accidents, etc.

En 2006, une étude du gouvernement canadien portant sur neuf grands centres urbains indique que le coût de la congestion récurrente dans ces zones urbaines varie entre 2,3 et 3,7 milliards de dollars par an [69]. Ce coût est réparti de la manière suivante :

- a. 90 % sont attribués au temps perdu par les conducteurs et les passagers dans la circulation ;

- b. 7 % sont dus à l'augmentation de la consommation de carburant ;
- c. 3 % représentent l'augmentation des émissions de gaz à effet de serre.

Il faut noter qu'en 2012, une autre étude a montré que le coût de la congestion dans ces mêmes zones était passé à 4,6 milliards de dollars [7].

Un exemple important de perturbation est l'effondrement du pont I-35W de la rivière Mississippi à Minneapolis (USA) qui s'est produit le 1^{er} août 2007 et qui a causé des morts et des blessés [6]. Cet incident a déstabilisé complètement la circulation et a entraîné un changement d'itinéraire et d'horaire de départ chez de multiples usagers. Il a fallu six semaines pour que la circulation routière se stabilise. Ce genre d'évènement perturbe non seulement la circulation, mais produit aussi un impact direct sur les usagers. Ceci est un exemple du choc psychologique qui se manifeste par du stress, car les gens doivent changer leur parcours habituel (pour se rendre au travail par exemple), changer les heures usuelles de départ, etc.

Ces impacts peuvent être minimisés si l'on anticipe l'éventualité d'un tel incident. D'après la recherche de Zhu et al. [6], un plan de contournement rapide et bien conçu peut contribuer à améliorer l'état du trafic en aidant les usagers à suivre un trajet alternatif. Au vu de ce constat, l'on peut soutenir qu'une maintenance et une analyse du réseau routier sont nécessaires. Celles-ci auront pour but à la fois de rassembler les informations sur les infrastructures et les incidents, et de faciliter l'accès à ces informations afin de réaliser la maintenance, d'une part, et de trouver des solutions aux problèmes de routage d'autre part. De plus, l'état vétuste des infrastructures routières de la province de Québec exige une attention particulière. L'absence de prise de mesures adaptées peut nuire à la qualité de la circulation routière et même provoquer un arrêt complet de cette circulation.

Parmi les solutions envisageables pour mieux gérer et entretenir les infrastructures routières, les Systèmes d'Information Géographique (SIG) semblent être un choix de premier ordre. Ils représentent de façon précise les objets géographiques en utilisant des données spatiales. Une variante de SIG appelée SIG Web ou WebGIS permet l'accès à ces objets via le Web. Il existe plusieurs projets de SIG qui ont été développés à travers le monde et qui ont prouvé leur utilité, pour résoudre diverses problématiques dans des domaines tels que le transport, les télécommunications, le tourisme, l'hydrologie, la protection civile, etc. Nous allons donc privilégier l'usage d'un SIG Web pour notre projet.

Au Québec il n'existe aucun système de cartographie Web pour la gestion des infrastructures routières relevant du génie civil en particulier, du moins les initiateurs de ce projet ne disposent pas d'un tel outil. Par ailleurs, aucun des SIG qui ont été développés n'est dédié à la gestion des infrastructures routières et au calcul d'itinéraires se basant sur les incidents liés aux ponts tels que les effondrements.

Objectifs de ce travail

C'est dans cette optique que nous menons cette étude dont l'objectif général est de simplifier la maintenance des infrastructures routières de la province de Québec en facilitant l'accès aux informations inhérentes à celles-ci, et de répondre au souci d'assurer un service de circulation routière continue quand il y a un effondrement de pont.

Les objectifs spécifiques de ce projet de recherche sont donc :

- d'établir un système d'information cartographique détaillé portant sur les infrastructures routières de Québec, interrogeable par de simples sélections des objets représentés sur la carte.
- de mettre en place un module d'analyse et d'optimisation proposant des chemins alternatifs au trafic en cas de défaillance d'une partie du réseau routier.

Hypothèses

Pour ce travail, il est supposé qu'un système Web d'information géographique interactif va aider les professionnels en génie civil à avoir une perception globale et précise de leur patrimoine d'infrastructures. Il doit faciliter la gestion de ces dernières et assurer un service continu, et ce, même quand il survient un incident susceptible de mettre hors d'état d'usage une partie du réseau routier. Nous croyons pouvoir réduire les conséquences des perturbations résultantes des incidents sur le trafic routier.

Nos hypothèses spécifiques sont :

- que l'on peut gérer les données liées aux caractéristiques et à la géolocalisation du réseau routier ainsi que les infrastructures connexes à ce dernier dans une base de données relationnelle à référence spatiale (permettant de manipuler des données géographiques).
- que la gestion des infrastructures routières et de calcul d'itinéraires par des interactions des usagers avec une carte géographique électronique serait d'une simplicité et d'une efficacité notoire.

Plan du mémoire

Le mémoire est organisé comme suit. L'introduction présente la problématique, les hypothèses utilisées ainsi que les objectifs. Au chapitre I, nous définissons les différents concepts nécessaires à la bonne lecture de ce mémoire. Le chapitre II présente une revue de littérature sur les différentes technologies liées à la gestion d'infrastructures routières, au calcul d'itinéraires et aux applications cartographiques Web. Aux chapitres III et IV, nous exposons la solution proposée et le processus suivi lors de sa réalisation. Le chapitre V présente les résultats. Enfin, nous concluons notre travail et dégageons des perspectives.

Chapitre 1 Notions

Pour bien comprendre les tenants et les aboutissants de ce projet, il est nécessaire de se familiariser avec les domaines de la géomatique, de la cartographie et des graphes. Dans cette section nous présentons les notions utiles à la compréhension de ce travail.

Nous présentons notamment les concepts d'information géographique et de cartographie qui sont primordiaux à la compréhension de l'information manipulée par un SIG. Ensuite nous présentons les concepts reliés directement au SIG et au SIG Web. Ainsi, après avoir défini les concepts de SIG, nous abordons la question de l'échange d'information. En effet, les SIG ont besoin d'échanger de l'information avec des clients ou d'autres SIG, d'où la nécessité d'utiliser la norme Open Geospatial Consortium (OGC) qui facilite cette communication grâce aux services « Web Feature Service (WFS) » et « Web Map Service (WMS) ». Enfin, comme notre travail concerne également le calcul d'itinéraires basé sur des algorithmes de parcours de graphes, nous présentons le concept de graphes ainsi que les algorithmes utiles pour les graphes.

1.1 Information géographique

Pour produire des cartes, nous avons nécessairement besoin des informations géographiques. L'information géographique décrit un objet (par exemple, un pont ou une route), un phénomène ou encore une action du monde réel [65]. Elle comprend trois propriétés [64] :

- **les propriétés non référencées** (ou non localisées) :

Ces propriétés sont celles qui décrivent l'entité par le biais de ses attributs. Les valeurs de ces propriétés sont appelées données attributaires. Elles sont indépendantes de la localisation. Si l'on considère un pont par exemple, elles peuvent être le nom, la longueur, la hauteur, le poids supporté, etc.

- **les propriétés référencées :**

Ces propriétés sont les attributs décrivant la localisation (par exemple, les coordonnées d'un pont), les formes géométriques (par exemple, les lignes représentant une route) et la relation existante entre ces entités. La relation réfère à la connectivité des entités entre elles, la notion d'adjacence par exemple, etc. Bref ce sont les données géométriques.

- **les propriétés temporelles :**

Ces propriétés sont celles par lesquelles on observe les changements de propriétés de l'objet d'étude dans le temps. Le suivi de ces changements se base notamment sur ses attributs localisés et non localisés.

1.2 La cartographie

La cartographie concerne toute la démarche de la création de cartes. Selon [61], elle a pour but la représentation de la terre ou d'une autre planète sous une forme géométrique et graphique grâce à la conception, la préparation et la réalisation de cartes. La cartographie est à la fois une science, un art et une technique.

1.2.1 La carte

Une carte est une représentation géométrique, plane, simplifiée et conventionnelle de toute ou d'une partie de la surface terrestre, et cela, dans un rapport de similitude convenable qu'on appelle échelle [60]. Une carte est un objet ancien, qui présente plusieurs facettes du monde réel et qui fait l'objet de multiples utilisations dépendamment du domaine considéré. Cependant, toutes les cartes ont un point commun qui est la représentation d'une partie de la surface terrestre.

Les cartes sont importantes parce que, grâce à cette représentation de l'espace terrestre et surtout la représentation numérique de celui-ci, on peut mener des études sur des phénomènes réels et ainsi pouvoir leur appliquer les sciences et techniques théoriques développées, comme les techniques de la théorie des graphes notamment.

La cartographie concerne non seulement la production de cartes, mais aussi la localisation d'objets représentés sur la carte. Pour cela, il existe plusieurs systèmes de référence : le système de référence terrestre (SRT) et de coordonnées planimétriques, le système de référence vertical (SRV) et l'altitude, et le système de coordonnées de référence (SCR) que nous utilisons.

1.2.2 Système de Coordonnées de Référence (SCR)

Pour faire l'étude des phénomènes se déroulant sur la terre via les SIG, on doit préciser leurs emplacements, c'est-à-dire les situer sur la terre. On parle alors de géoréférencement, géopositionnement, géolocalisation, etc. En d'autres termes, cela signifie assigner des coordonnées à l'objet ou à l'évènement en question. Pour ce faire, il faut un repère. Grâce aux systèmes de

coordonnées de référence (SCR), chaque emplacement sur la terre peut être spécifié par des coordonnées de points. Les SCR sont en général issus de deux groupes, soit les SCR projetés et les SCR géographiques.

Les **SCR projetés** utilisent la latitude et la longitude (en degré) et, quelquefois, une valeur d'altitude. La norme la plus populaire parmi celles-ci est le système géodésique mondial (WGS84), mis au point par le Département américain de la défense en 1984.

Les **SCR géographiques** sont définis dans un plan à deux dimensions et la localisation d'un point se base sur deux coordonnées en X (axe horizontal) et en Y (axe vertical). Si le système est à trois dimensions, alors il y aura un troisième axe, le plus souvent nommé Z, qui constitue la troisième dimension de l'espace. Dans ce cas, un point est défini par trois coordonnées.

Concernant la **Projection de cartes**, comme la terre n'est pas plate et que les personnes utilisent dans leur quotidien soit du papier ou des écrans qui sont tous plats, il est nécessaire d'avoir recours à la projection. Celle-ci est un processus consistant à projeter la terre de sa forme réelle (ellipsoïdale) vers une forme aplatie sur un support à deux dimensions.

1.3 Systèmes d'information géographiques (SIG)

Dans la littérature plusieurs définitions ont été données aux Systèmes d'information géographiques (SIG). Nous en présentons trois qui ont servi à établir la définition que nous utilisons dans ce travail. D'après Jiang et Zhang [71], le SIG est un système de collecte et de stockage de données qui permet de réaliser une analyse intégrante des informations indépendantes au sein d'une même carte facilement compréhensible, d'effectuer des analyses complexes et de présenter les résultats visuellement sous forme de cartes, de tableaux ou de graphiques.

Selon McHarg [66], « un SIG est un système d'information spatiale pour lequel la terre est l'objet de référence fondamental. Il traite un ensemble de données repérées dans l'espace, structuré de façon à pouvoir tirer des synthèses utiles à la décision ».

Roberts et Hadjiconstantinou [61] définissent le SIG comme étant « un ensemble de principes, de méthodes, d'instruments et de données à référence spatiale utilisées pour saisir, conserver, extraire, mesurer, transformer, analyser, simuler et cartographier les phénomènes et les processus distribués dans l'espace géographique ».

Pour nous, un SIG est un ensemble d'éléments (données à référence spatiale ou non, équipements, logiciels, procédures, personnels, institution, etc.) organisés globalement et qui se coordonnent (mesure, saisie, stockage, traitement) pour concourir à un résultat (restitution de l'information géographique) facilitant les prises de décisions éclairées. Par extension, un SIG Web (WebGIS) est une technologie permettant l'accessibilité aux fonctionnalités d'un SIG via le Web. Zhang et al [46] et Jia et Han [55] décrivent le SIG Web comme étant simplement une technologie issue d'une symbiose de la technologie internet (Web) et du développement du SIG. Ainsi, cette technologie Web rend les fonctionnalités classiques du SIG, en particulier les informations géographiques, accessibles sur internet, donc accessibles à toute personne connectée [55].

Les SIG sont une technologie mature qui a fait ses preuves. Ils sont capables de traiter de multiples sujets provenant de domaines variés tels que le transport, l'énergie, l'environnement, etc. Ils peuvent aussi représenter et gérer des informations géographiques, c'est-à-dire des informations représentant les objets physiques situés sur la terre. Cette représentation facilite les analyses afin d'appréhender la réalité.

La thématique de ce mémoire fait appel aux SIG, mais principalement pour les infrastructures de réseaux routiers.

1.3.1 Formats des données

Le SIG se base sur l'information géographique. Il est donc important de sauvegarder cette information de manière à pouvoir l'utiliser lors des analyses. Or pour faire une bonne analyse, il est nécessaire de disposer de données bien structurées. Il existe actuellement deux formats de stockage classiques pour les SIG [64], les formats raster et vecteur.

1.3.1.1 Format raster

Le format raster (image) représente l'étendue géographique par une grille (matrice ligne/colonne) divisée en cellules appelées pixels qui sont les plus petites unités dans le format raster. Une valeur (couleur) est associée à chaque pixel. Pour avoir une haute résolution, il faut représenter l'image avec le plus de pixels possible. Cela va de pair avec le coût de stockage.

Le format raster présente des avantages tels qu'un coût de saisie peu onéreux, un traitement rapide des requêtes et une représentation visuelle conviviale et intuitive pour les usagers.

Il a pour inconvénients d'être coûteux en matière d'espace de stockage et lourd en manipulation. De plus, comme tous les objets (pont Champlain, rue Laurier, etc.) sont indissociables, on ne peut manipuler chaque objet formant la carte raster individuellement. Par exemple, il est impossible de colorer le pont « x » de la carte raster en rouge et le pont « y » en vert. Le principal usage du format raster est de représenter des images qu'on ne veut pas modifier. Par exemple, nous représentons l'arrière-plan de notre application, représentant la zone d'étude (carte de la ville de Québec), dans un format raster.

1.3.1.2 Format vecteur

a) Description

Le format vecteur se base sur le concept d'objets géométriques, c'est-à-dire chaque information géographique est représentée par un objet indépendant muni de caractéristiques. Les éléments de base permettant la représentation de toute entité sont le point, pour représenter une information géographique, la ligne, pour représenter un ensemble de segments continus dont chaque segment est représenté par deux points, et le polygone pour représenter les entités complexes combinant les points et les lignes. Par exemple, pour représenter un réseau routier, dans le but de faire une analyse telle que le calcul d'itinéraire, l'utilisation du format vecteur est nécessaire.

Ce format possède l'avantage :

- d'occuper moins d'espace mémoire ;
- de représenter chaque objet de l'image individuellement ;
- de faciliter l'exploitation des données ;
- d'être précis sur les opérations (par exemple, il manipule les coordonnées de l'image contenue dans une cellule).

Les inconvénients de ce format résident dans une numérisation longue et une exécution lente des requêtes. Toutefois, il reste possible de passer du format vecteur au format raster et inversement.

L'un des formats vecteurs les plus utilisés dans le monde des SIG est le format shapefile. Nous avons reçu la majorité de nos données en format shapefile et d'autres ont été converties en shapefile avant de les importer dans la base de données.

b) Format Shapefile

Le format shapefile est un format vecteur qui est très utilisé par les SIG. Initialement, il a été développé par *Environmental systems research institute* (ESRI), pour devenir par la suite un standard dans le monde des SIG.

Un fichier en format shapefile a pour extension .shp et doit toujours être accompagné par deux autres fichiers (voir Figure 1) du même nom, mais avec des types différents : un fichier .dbf, contenant les données sur les attributs des objets contenus dans le shapefile, et un fichier .shx qui contient l'index des enregistrements du fichier shapefile (objets géométriques). Il peut aussi être accompagné d'autres fichiers fournissant des informations complémentaires telles que :

- les index spatiaux des formes qui n'apparaissent qu'après une requête ou une jointure (fichiers de type .sbn et .sbx),
- les index spatiaux des formes pour les shapefiles en lecture seule (fichiers de type .fbn et .fbx),
- les index des attributs des champs actifs d'une table (fichiers de type .ain et .aih),
- l'information sur le système de coordonnées (fichiers de type .prj),
- les renseignements sur les métadonnées du fichier shapefile (fichiers de type .xml),
- les informations sur la légende (fichiers de type .avl).

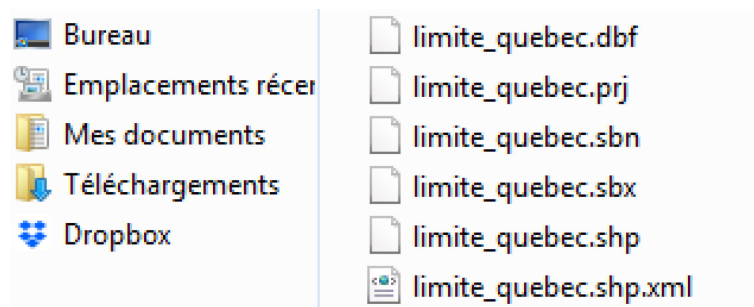


Figure 1 - Image d'un format shapefile muni des fichiers complémentaires, capturée dans Windows

1.3.2 Les couches

Une carte est formée par plusieurs couches. Le concept de couches est un moyen de représenter la géographie d'un lieu. Il permet de représenter les objets similaires ensemble dans une même couche pour mieux les gérer. Une couche est donc un ensemble d'entités géographiques de même nature qui décrit comment ces entités sont réparties et représentées sur un espace géographique. Par exemple, les ponts peuvent être représentés par une couche et les routes par une autre couche. Les éléments importants d'une couche sont les jeux de données référencées, les symboles cartographiques de représentation et les étiquettes textuelles (propriétés). La notion de superposition est une technique consistant à placer une couche au-dessus de l'autre. La figure 2 illustre cette superposition de couches. Ce processus peut s'appliquer sur plus de deux couches. Elle permet surtout d'analyser la relation existante entre les éléments des différentes couches afin de tirer des conclusions. Timlinson [72], un des pionniers des SIG considère le concept de couches et de la superposition de couches comme une des pierres angulaires des SIG.

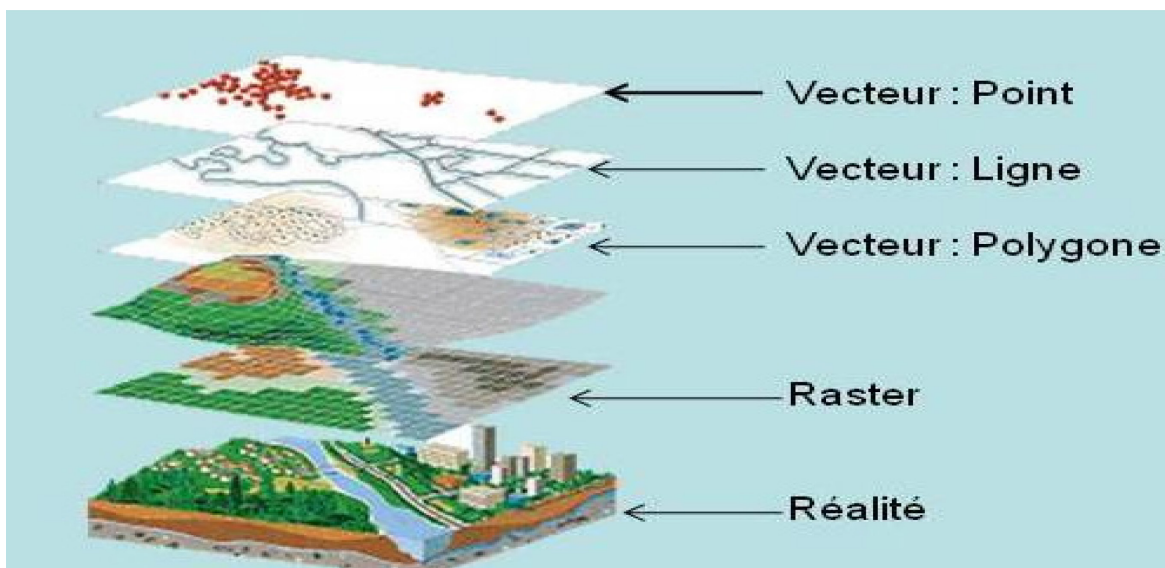


Figure 2 - Superposition des couches

Source : Institut français de l'éducation de Lyon (l'IFÉ/ENS), France.

1.3.3 Normes Open Geospatial Consortium (OGC)

Afin que les SIG puissent communiquer entre eux ou que les clients Web puissent interagir facilement avec un SIG, il est nécessaire d'établir une norme pour les services fournis par les SIG.

L'Open Geospatial Consortium (OGC) est une organisation à but non lucratif fondée en 1994 dont l'objectif était de mettre en place un standard international ouvert assurant l'interopérabilité des données géospatiales pour ainsi faciliter la coopération entre les acteurs dans le domaine de SIG.

Les standards établis dans le domaine de données géoréférencées, fournissent des documents techniques détaillant l'encodage des données et les interfaces (dispositifs permettant l'accès au service d'un système par un autre système). Ces documents servent de feuille de route aux concepteurs pendant le processus de développement pour faire des interfaces ouvertes et des encodages normés afin de garantir une interopérabilité entre les différents produits des concepteurs.

1.3.3.1 Services

Les serveurs suivant la norme OGC, tel GeoServer, fournissent des services spécifiés par les protocoles du standard **OGC**. Parmi ces services on peut citer : « Web Feature Service » et « Web Map Service ».

a) Web Map Service (WMS) [43]

WMS est un protocole donnant accès aux images (raster) géoréférencées produites à partir des données de type vecteur ou raster à des clients distants. L'interface fournie sous le service WMS est accessible via le protocole HTTP.

Un serveur proposant WMS produit dynamiquement une carte de données géographiques géoréférencées dans un format image tel que JPEG, PNG, GIF, etc. Il faut noter que c'est l'utilisateur qui précise le type du format dans sa requête au serveur. Il peut formuler une requête URL, munie des attributs et valeurs, à l'endroit du serveur à travers un navigateur Web standard. Le contenu de l'URL varie en fonction de l'opération effectuée. Quand il s'agit d'une carte, par exemple, l'URL contiendra :

- les informations qui doivent être affichées sur la carte,
- la délimitation de la partie de la Terre à cartographier,
- le système de coordonnées de référence (SCR),
- la largeur et la hauteur de l'image de sortie.

Une requête dans le format WMS permet aussi d'extraire :

- des métadonnées sur le service proposé, par exemple les métadonnées de la couche de ponts sont les noms des attributs;
- des informations sur les objets affichés sur la carte.

La figure 3 montre un exemple de requête WMS qui comprend dans l'ordre :

- l'adresse du serveur et le type de service (<http://localhost:8080/geoserver/gestionIncidentRoute/WMS?SERVICE=WMS>),
- la version du service WMS, selon le standard international (*VERSION=1.3.0*),
- le type de la requête. Celle-ci permet de télécharger la carte. La réponse est un fichier image (*REQUEST=GetMap*),
- le format de l'image souhaité en réponse (*FORMAT=image%2Fpng*),
- le système de référence ; ici il s'agit de EPSG:3857 (*CRS=EPSG%3A3857*),
- les différents styles des couches des cartes demandées. Pour cette requête le style est celui par défaut du serveur (*STYLES=*),
- les dimensions en pixel de l'image à construire (*WIDTH=2024&HEIGHT=1004*),
- les limites en « x » et en « y » de la partie de la couche à afficher (*BBOX=-79377725.023787958%2C5986969942878734%2C7918391.232946076%2C5916558.012614256*).

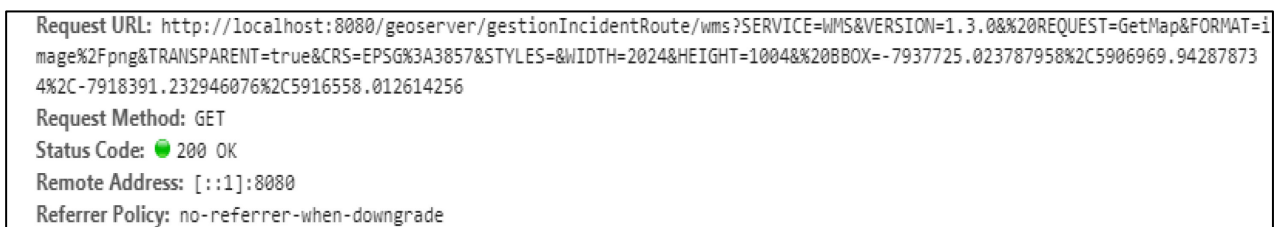


Figure 3 - Capture d'une requête WMS

b) Web Feature Service (WFS) [63]

Le service WFS propose un accès aux données dans le format vecteur géoréférencées. Les utilisateurs peuvent télécharger les données par une simple requête HTTP au moyen d'une URL formatée. Plutôt que de produire une carte, comme le fait WMS, WFS permet d'accéder directement aux objets spatiaux et de les manipuler. Il est donc plus optimal et pratique.

Le serveur fournissant le service WFS utilise le langage GML (Geography Markup Language) qui est un langage dérivé de XML pour encoder, manipuler et échanger les objets spatiaux.

La figure 4 représente un exemple de requête WFS qui comprend dans l'ordre :

- l'adresse du serveur et le type de service WFS (<http://localhost:8080/geoserver/gestionIncidentRoute/OWS?SERVICE=WFS>),
- la version du service WFS, selon le standard international (*VERSION=1.0.0*),
- le type de l'opération. La valeur « *GetFeature* » permet d'obtenir les instances des objets géographiques (*REQUEST=GetFeature*),
- la couche d'objets dont on désire obtenir la description (*typename = gestionIncidentRoute : rtss_rvq*),
- le format dans lequel est fournie la description des objets (GML par défaut), ici le format demandé est json (*outputFormat=application/json*),
- le système de référence désiré pour les géométries des objets retournés, ici le système de référence est EPSG:4326 (*srcname = EPSG:4326*),
- le rectangle qui délimite l'espace concerné par la requête (*BBOX=%27%20+%20extent.join(%27,%27)%20=%20%27, EPSG : 3857%27*).

```
Request URL: http://localhost:8080/geoserver/gestionIncidentRoute/ows?service=WFS&%27%20+%20%27version=1.0.0&request=GetFeature&typename=gestionIncidentRoute:rtss_rvq&%27%20+%20%27outputFormat=application/json&srsname=EPSG:4326&%27%20+%20%27bbox=%27%20+%20extent.join(%27,%27)%20+%20%27, EPSG:3857%27;
Request Method: GET
Status Code: 200 OK
Remote Address: [::1]:8080
Referrer Policy: no-referrer-when-downgrade
```

Figure 4 - Capture d'une requête WFS

1.3.4 Client Web (OpenLayers)

OpenLayers est une bibliothèque pour aider au développement d'applications clientes de cartographie pour le Web. Elle a vu le jour en 2006 ; le code source est libre, mais elle est devenue populaire en 2007 lorsque la fondation OpenStreetMap l'a adoptée pour son site Web. Aujourd'hui, OpenLayers est probablement la bibliothèque la plus puissante dans le monde des logiciels libres pour le développement d'applications de cartographie Web. Elle a les fonctions nécessaires pour manipuler les informations géographiques.

Au début, la bibliothèque OpenLayers était conçue pour visualiser les données vectorielles et les données rasters. Aujourd'hui elle permet :

- la lecture et l'écriture de données dans plusieurs formats dont Geography Markup Language (GML), GeoJSON, Keyhole Markup Language (KML) ;
- l'utilisation de données ayant différents systèmes de projection ;
- le stockage et la manipulation de différents types de données ;
- la mise en forme des objets en se basant sur les attributs. Par exemple, colorer certains ponts en orange et d'autres en vert [67].

1.4 Les graphes

De nombreux problèmes sont difficiles à résoudre dans leur état originel. Or le plus souvent, ils sont traduisibles sous forme de graphe. Il existe plusieurs algorithmes permettant de calculer un chemin dans un graphe. Dès lors, la solution va consister à rechercher un nœud spécifique dans le graphe.

Un graphe est représenté par un ensemble de nœuds a_1, a_2, \dots, a_m reliés entre eux par un ensemble de lignes v_1, v_2, \dots, v_n appelées arcs. Un nœud est un point qui représente le sommet dans un réseau et un arc (arête si le graphe est non orienté) sert à relier deux sommets.

Formellement, un graphe fini $G = (V, A)$ est défini par l'ensemble fini $V = \{v_1, v_2, \dots, v_n\}$ dont les éléments sont appelés sommets et par l'ensemble fini $A = \{a_1, a_2, \dots, a_m\}$ dont les éléments sont nommés arcs. On associe un poids à chaque arc $(a_n, a_m) \in A$. Ce poids peut être soit une longueur supérieure à zéro ou une estimation du temps de parcours de l'arc. La longueur d'un chemin (distance séparant deux sommets) dans un graphe est la somme des longueurs de ses arcs.

1.4.1 Recherche dans un graphe

Dans notre projet, nous devons parcourir le graphe représentant le réseau routier afin d'effectuer le calcul d'itinéraire. Le principe de parcours d'un graphe consiste à partir d'un sommet et à explorer tous les autres sommets du graphe de proche en proche. Par exemple, dans le cas d'un site Web représenté par un graphe, les pages vont jouer le rôle de sommets, c'est-à-dire qu'elles feront l'objet de visite, et les arcs, celui des liens entre les pages permettant la navigation d'une page à l'autre. Il existe deux techniques de parcours de graphe : parcours en largeur et parcours en profondeur. Il

faut noter que le graphe est transformé en arbre couvrant (arbre inclus dans le graphe et connectant tous les sommets du graphe) sur lequel les deux parcours sont effectués.

1.4.1.1 Parcours en largeur

Pour effectuer le parcours en largeur, on commence par balayer tout d'abord les sommets situés au premier niveau juste après celui de départ. Une fois que les sommets du premier niveau sont tous visités, on passe aux sommets du deuxième niveau et cela se répète jusqu'à la fin. Du point de vue technique, on peut utiliser la structure de données informatique **FILE** pour implémenter le parcours en largeur. Le principe de cette structure est que les premiers éléments ajoutés à la **FILE** sont les premiers à en être retirés. La Figure 5 fournit une illustration de ce type parcours.

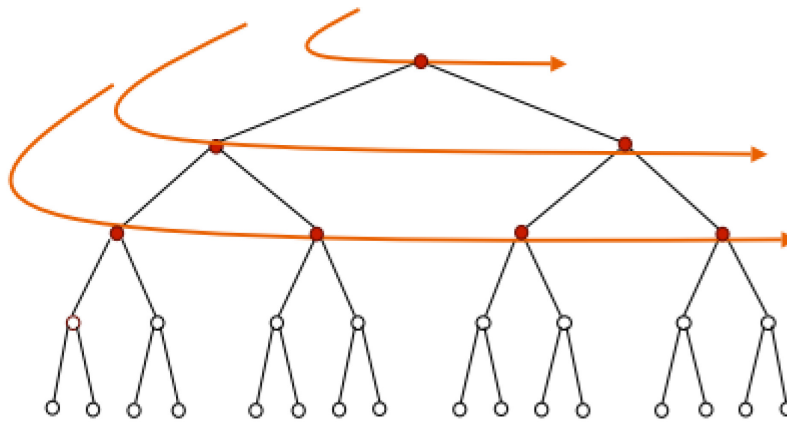


Figure 5 - Parcours en largeur (figure tirée du Cours IA A. Cornuéjols)

1.4.1.2 Parcours en profondeur

Pour effectuer le parcours en profondeur, on commence par le sommet et l'on suit un même chemin sans relâche jusqu'à la feuille. Une fois la feuille atteinte, on recommence à partir du sommet adjacent supérieur à la feuille atteinte, si ce sommet mène à d'autres feuilles non visitées, sinon on remonte au sommet voisin supérieur. On effectue récursivement ce processus jusqu'à atteindre toutes les feuilles. La Figure 6 illustre ce parcours. On peut utiliser la structure de données **PILE** pour implémenter le parcours en profondeur. Dans le principe de cette structure, le dernier élément inséré est le premier à être retiré. Cette structure est moins gourmande en espace mémoire.



Figure 6 - Parcours en profondeur (figure tirée du Cours IA A. Cornuéjols)

Chapitre 2 Etat de l'art

Notre travail se fait à la frontière de trois domaines, soit les SIG, le génie civil et la recherche opérationnelle. Des multiples travaux ont été menés sur ces domaines individuellement ou ensemble. Dans ce chapitre, nous présentons les travaux pertinents afin de faire ressortir l'originalité de notre travail.

2.1 Gestion des infrastructures routières

Les infrastructures routières jouent un rôle important dans l'économie d'une ville ainsi qu'au plan social. Une bonne méthode de gestion peut préserver ces infrastructures de dégradations prématurées. Cela nécessite de conserver beaucoup d'informations de maintenance. Étant donné que les données utilisées dans le processus de prise de décision ont des composantes géographiques, l'utilisation des technologies géographiques (spatiales) semble être une bonne solution. Les Systèmes d'information géographique (SIG) sont capables d'intégrer des données volumineuses et géographiques, et de simplifier l'utilisation de ces données pour mieux gérer les infrastructures routières.

2.1.1 Les données

2.1.1.1 Données pour l'identification des infrastructures routières

a) Introduction

Jiang et Zhang [45] estiment que 80 % des données existantes dans le monde sont géographiques (c'est-à-dire qu'elles ont des composantes géographiques). En d'autres termes, ces données peuvent être représentées par une carte via un SIG. Cela veut dire que les données de gestion des réseaux routiers ne constituent pas une exception à la règle, car elles sont en majorité géographiques.

En se fiant à la littérature, généralement dans le cadre des projets de gestion des infrastructures routières, les données manipulées sont divisées en deux catégories. Il s'agit de données statiques et dynamiques [47].

Les données statiques sont liées à la description de l'identité et à des propriétés de l'entité faisant l'objet de l'étude (pont, chaussée, ponceaux, tunnels, etc.). En général, ce type de données ne

varient pas pendant des longues périodes. Des exemples de ce type de données sont la géométrie des routes, la structure de la chaussée, l'historique d'entretien des routes, la charge axiale supportée par un pont, les installations de protection des ponts et le sol de fondation.

Les données dynamiques, dites opérationnelles, dépendent du temps et des opérations. Dans la plupart des cas, on se réfère à ces données pour constater les changements et effectuer des analyses afin de prendre les décisions. Des exemples de telles données sont fournis par He et Li [47], à savoir les indices d'évaluations globales qui sont décrits à la section « modèles mathématiques d'évaluation » et l'ensemble de données sur la dégradation de la chaussée (selon le type et le degré de dommages, le poids, etc.).

Dans cette section, nous introduisons des données communes de gestion de maintenance de chaussées, de ponts et de tunnels, puis nous présentons les données utilisées lors de la maintenance individuelle d'une chaussée, d'un pont ou d'un tunnel. Nous présentons aussi un exemple d'un plan pratique et détaillé d'une inspection de maintenance d'un pont. Enfin, nous parlons des modèles mathématiques calculant des indices d'évaluations globales d'une chaussée, d'un pont et d'un tunnel qui déterminent les états de ces ouvrages d'art.

b) Données nécessaires pour la gestion des informations sur la maintenance

Dans la littérature, plusieurs articles traitent de la gestion de maintenance des infrastructures routières en se basant sur des éléments tels que les distorsions, les orniérages, l'excès d'asphalte, les fissures (de bord, faïençage, transversales, longitudinales et aléatoires ou en blocs), les détresses mixtes, les failles de surface [48], le nivellement, les dommages de chaussée et sa résistance, les tests antidérapants [42], l'âge, le trafic routier, la ventilation, l'hydrologie, la géologie, la profondeur et l'imperméabilité [43].

En plus de ces éléments, il faut vérifier ([47] [49] [44]) :

- la rugosité de la chaussée,
- l'état du joint de dilatation,
- les installations (d'alimentation électrique, de drainage, d'éclairage, de sauvetage et de ventilation),
- le niveau de déformation du pont,
- le coefficient de la charge du vent,

- la vulnérabilité sismique,
- le type de structure,
- le niveau de déformation du mur de soutènement,
- le niveau de rigidité et de vibrations du pont,
- le type de fondation,
- le calcul de la charge externe,
- le niveau détérioration des matériaux.

Cependant, les éléments énumérés ci-dessus ne sont pas tous utilisés en même temps. Chaque projet utilise les paramètres qui lui sont nécessaires (voir Tableau 1).

Tableau 1 - Données pour la maintenance d'un pont, d'un tunnel et d'une chaussée

Projet	Paramètres	Source
Maintenance d'une Chaussée	Rugosité de la chaussée, test anti-dérapage, nivellement, résistance de la chaussée, orniérage, dommages de la chaussée, fissuration, etc.	[49] [42]
Maintenance d'un pont	Chaussée du tablier du pont, installations de drainage, longueur, largeur, joint de dilatation, etc.	[8]
Maintenance d'un Tunnel	Coordonnées, longueur, largeur, profondeur, type des matériaux, trafic, température, humidité, atmosphère (composition chimique), eaux souterraines, ventilation, date d'inspection, fissures, etc.	[43]

Il faut noter que la gestion des infrastructures routières est nécessaire dans différents domaines, comme la maintenance, le calcul et l'optimisation des trajets, etc. Cependant, selon le projet d'étude, certains éléments peuvent être plus prépondérants que d'autres. C'est-à-dire que le niveau d'importance d'un élément quelconque dépendra du projet considéré. Par exemple, considérant le tablier d'un pont, notons que [8] :

- la largeur du tablier affecte la capacité du trafic (un élément important dans un projet d'écoulement de trafic),

- la détérioration du tablier affecte la capacité du pont (un élément important dans un projet de transit de gros camions),
- l'épaisseur du tablier affecte le coût, étant donné que la quantité de matériaux utilisée dépend de l'épaisseur (un élément important pour un projet d'allègement du coût de construction).

C'est ainsi que le tableau 1 fournit un classement d'éléments utilisés selon le projet, soit la gestion de maintenance d'une chaussée, d'un pont et d'un tunnel.

2.1.1.2 Données d'inspection d'un pont pour la maintenance

Afin de gérer la maintenance d'un pont, il faut disposer d'un plan identifiant tous les éléments à inspecter. Ce sont les éléments critiques dans la sécurité d'un pont. C'est ainsi que Shintoku et al. [8] ont proposé une classification pour les données d'inspection d'un pont dans le cadre de la maintenance. Toutefois, ils ont signalé le manque de données sismiques dans le cadre de la gestion des ponts. Ils suggèrent dans ce cas un recours à la simulation, si l'on dispose d'une base de données sismique. Dans l'énumération qui suit, nous présentons les classes proposées par Shintoku et al. [8] :

a) Identification et caractérisation du pont

Pour évaluer un pont, on recommande d'identifier les éléments suivants :

- longueur du pont,
- largeur du pont,
- type de fondation,
- localisation du pont,
- date de construction,
- volume du trafic attendu.

b) Inspection de la superstructure / infrastructure

Les données utilisées pour l'inspection à ce niveau regroupent :

- les dommages de chaque élément constituant le pont selon le type et les matériaux,

- l'identification des éléments endommagés par leurs noms et les mesures préconisées contre les dommages enregistrés,
- le nombre d'éléments endommagés avec leurs noms et leurs types.

c) Données sur le niveau de détérioration

Ces données traitent de l'évaluation de l'état du pont et de ses composantes ainsi que des besoins de maintenance de chaque élément de la superstructure et de l'infrastructure.

d) Données de l'historique de maintenance

Ces données comprennent l'année et la méthode utilisée pour les maintenances majeures, le coût, etc.

2.1.1.3 Indicateurs de détérioration

Le processus d'obtention des données citées à la section précédente est très fastidieux. En effet, pour inspecter les infrastructures routières, il faut récolter les informations sur le terrain, et ceci, à chaque évaluation. Or, il existe des modèles, développés pour prédire la détérioration des infrastructures routières, qui évitent le processus répétitif de récolte des données.

a) Modèle d'évaluation d'une chaussée

Ce modèle d'évaluation de performance de la chaussée se base essentiellement sur des éléments liés à la fondation et au bitume de la route. Les indicateurs de base qui servent au modèle mathématique pour évaluer les dommages de la route sont :

- l'indice de dégradation de la chaussée (PCI) ;
- l'indice de la rugosité de la chaussée (RQI) ;
- l'indice de l'orniérage (RDI) ;
- l'indice de résistance au dérapage (SRI).

Les valeurs PCI, RQI, RDI et SRI sont obtenues principalement par des mesures in situ par un équipement muni de lasers à bord d'un véhicule dédié, et ce, à la vitesse de la circulation.

Avec ces éléments il devient possible de calculer une valeur approximative de l'indice de l'évaluation globale de l'état de la chaussée, appelé « Pavement Quality Index (PQI) » qui va servir pour des prédictions ultérieures ([42], [44], [47], [49], [50], [51]).

Le modèle linéaire est construit de la manière suivante :

$$PQI = PCI * P_1 + RQI * P_2 + RDI * P_3 + SRI * P_4 \quad (3-1)$$

Les coefficients P_1 , P_2 , P_3 et P_4 désignent les poids affectés aux indicateurs (PCI, RQI, RDI et SRI).

Le PQI prend ses valeurs dans l'intervalle [0, 100]. Les interprétations de ces différentes valeurs sont présentées dans le Tableau 2.

Tableau 2 - Interprétations des valeurs de prédiction d'une chaussée

Evaluation	Excellent	Bien	Moyen	Mauvais	Pire
PQI	≥ 85	[70, 85]	[55,70]	[40, 55]	< 40

b) Modèle d'évaluation d'un pont

L'évaluation de l'état d'un pont concerne essentiellement les cinq éléments suivants :

- l'état de la chaussée du tablier du pont (R_1),
- la mesure de la largeur (R_2),
- l'état du trottoir du pont (R_3),
- l'état des installations auxiliaires (R_4),
- l'état des installations de drainage (R_5).

Le modèle mathématique de l'évaluation globale du pont est [49] :

$$D = 100 - \sum_{i=1}^n R_i W_i / 5 \quad (3-2)$$

La variable R_i représente l'échelle d'évaluation de la $i^{\text{ème}}$ structure prenant ses valeurs dans [1 à 5], et la variable W_i le poids qui lui est attribué. Les valeurs de l'indice d'évaluation globale D varient dans l'intervalle [0 à 100]. Les valeurs du D sont interprétées selon le Tableau 3.

Tableau 3 - Interprétations des valeurs de prédiction d'un pont

Evaluation	Bien	Moyen	Mauvais	Pire
D	≥ 88	≥ 60	[40, 60[< 40

c) Modèle d'évaluation d'un tunnel

Pour évaluer l'état d'un tunnel, on évalue d'abord l'état (qualité) de chacune de ses composantes individuellement. Les composantes à évaluer sont :

- les installations de bloc alimentation électrique,
- les installations d'éclairage,
- les dispositifs de surveillance,
- les installations de ventilation,
- les installations d'extinction d'incendie et de sauvetage,
- la chaussée du tunnel,
- les installations de drainage,
- le portail du tunnel,
- la décoration intérieure du tunnel.

L'indice de l'évaluation globale T_n d'un tunnel est donné par la formule [49] :

$$T_n = 100 - \sum_{i=1}^n S_i W_i / 5 - \sum_{i=1}^n E_i W_i / 5 \quad (3-3)$$

La variable S_i est l'échelle d'évaluation de la $i^{\text{ème}}$ structure prenant ses valeurs dans l'intervalle [1, 5], la variable E_i est l'échelle d'évaluation du $i^{\text{ème}}$ équipement électrique prenant ses valeurs dans l'intervalle [1, 5], et la variable W_i est le poids. L'indice de l'évaluation globale T_n puise ses valeurs dans l'intervalle [0, 100].

Les interprétations des valeurs de l'indice de l'évaluation globale T_n , sont celles fournies dans le Tableau 4.

Tableau 4 - Interprétations des valeurs de prédiction d'un tunnel

Evaluation	Bien	Moyen	Mauvais	Pire
T_n	≥ 90	[60, 88]	[40, 60]	< 40

2.1.1.4 Cause des détériorations des infrastructures routières

On retrouve dans la littérature plusieurs causes qui provoquent la détérioration des infrastructures routières. Il est possible de citer, par exemple, le vieillissement des infrastructures routières et l'augmentation du trafic routier ([51] [8]). En particulier pour la détérioration de la chaussée, Ibraheem et Falih [44], dans leur étude, ont développé un système de gestion de maintenance de la chaussée soulignant que la détérioration de la chaussée est due à différentes causes comme les intempéries, la vétusté de la chaussée, la charge de trafic, le mauvais entretien, etc.

D'autre part, Sandrone et Labiouse [43] ont identifié et analysé des pathologies causant la détérioration d'un tunnel à savoir les interactions chimiques du tunnel avec l'environnement avoisinant, l'utilisation de sels de déglacage, la pollution causée par le trafic (le gaz d'échappement et les bruits de véhicules) et la pollution produite par le système de ventilation (évacuation du gaz d'échappement et de la fumée en cas d'incendie en introduisant de l'air frais). À cela, s'ajoute la vétusté qui affecte les matériaux comme le béton (granulat, ciment, sable, etc.). Il y a aussi d'autres causes telles que l'infiltration de l'eau, la formation de la glace, la température, l'humidité, les incendies et les accidents de voiture.

2.1.1.5 Intérêt des SIG

Les SIG possèdent de nombreux avantages pour gérer les infrastructures routières. En raison de leur capacité à manipuler des cartes et à combiner toutes les informations relatives à l'entité étudiée et à son environnement, ils constituent un moyen adapté pour gérer des données d'un niveau de complexité élevé. Ils permettent ainsi de faire une analyse globale permettant une prise de décision rationnelle. Par exemple, Sandrone et Labiouse [43] ont étudié la structure des tunnels avec la masse rocheuse qui les entoure pour identifier les éléments causant leur détérioration. De plus, Ibraheem et Falih [44] signalent que les réseaux routiers s'étendent sur une vaste zone et interagissent avec divers éléments se trouvant dans leur entourage, y compris les rivières, les montagnes, les bâtiments et les autres routes. Ces études montrent l'omniprésence, dans les données utilisées par les systèmes permettant de gérer des réseaux routiers, des données géographiques intégrées dans un même système. C'est donc grâce aux SIG que la représentation et l'analyse des entités étudiées et leur environnement sont possibles.

De plus, les SIG permettent également une visualisation raffinée et un regroupement de données sous forme de couches. Pour Jiang et Zhang [45], chaque ensemble d'éléments homogènes forme une couche comme les ponts, les routes, les régions administratives, etc. Les SIG offrent aussi la possibilité de visualiser les données sous forme de graphes, de diagrammes, de tableaux, etc. Par exemple, avec ces possibilités de visualisation, il devient très aisé de suivre les données d'inspection et celles de la progression sur la construction.

Toutefois, il est à souligner que l'utilisation des SIG offre également des avantages économiques. Une bonne pratique de maintenance de routes à base d'un SIG va préserver ces dernières dans de bonnes conditions et ainsi maximiser le bénéfice économique et social. Un exemple réel de cette économie est illustré par l'étude faite sur un système de gestion des infrastructures intégré (*Integrated Infrastructure-asset Management System : IIMS [41]*). Dans cette étude, Lemer [41] a montré comment on peut améliorer le revenu public en faisant la gestion des infrastructures d'une région par le biais d'un SIG.

Toujours au sujet des avantages économiques qu'apportent les SIG, au cours d'une étude menée pour le compte de la coalition pour l'entretien et la réfection du réseau routier du Québec, Légraré [70] a montré que plus on retarde l'entretien d'une route, plus son entretien devient coûteux. Cette étude a montré que l'entretien d'un kilomètre de route de moins de 13 ans est évalué à somme de 1 000 \$, alors que la réfection d'un kilomètre de route dont l'âge varie entre 12 et 15 ans coûte 80 000 \$, soit 79 000 de plus que le premier cas. Finalement, la réhabilitation d'un kilomètre de route âgée de plus de 15 ans coûte 250 000 \$, soit 249 000 de plus que le premier cas.

2.1.2 Les technologies et l'architecture

Zhang et al [46] et Jia et Han [55] décrivent le SIG Web comme étant simplement une technologie issue d'une symbiose de la technologie internet (Web) et du développement d'un SIG. Ainsi, cette technologie Web rend les fonctionnalités classiques d'un SIG accessibles sur internet, dont le partage assuré par la publication des informations géographiques sur le web, donc accessibles à toute personne connectée [55].

2.1.2.1 Caractéristiques des SIG Web

Les SIG Web respectent des architectures précises et utilisent différentes technologies.

Parmi les technologies utilisées, nous avons entre autres Java/CORBA, DCOM/ActiveX et MapInfo MapXtreme. Le développement des SIG Web se base sur deux standards Java/CORBA et DCOM/ActiveX. Les applications développées selon ces deux standards s'exécutent sur diverses plateformes [55], car chaque application SIG classique est développée pour une plateforme dédiée.

Ainsi, vu que les applications s'exécutent sur diverses plateformes grâce à ces standards et qu'il y ait le besoin d'intégration de divers services (par exemple, le service de calcul d'itinéraire de pgRouting peut être fourni à GeoServer qui offre ce service à Openlayers sous format de service WFS) de SIG, il devient facile d'intégrer ces services via le Web [55], et ce, grâce à la standardisation des technologies Web libres par des organismes spécialisés comme l'Open Geospatial Consortium (OGC).

En ce qui a trait aux architectures utilisées par les SIG Web [55], Il y en a principalement trois :

a) Architecture centralisée

Dans cette architecture, l'information est stockée, parcourue, recherchée, récupérée et maintenue au niveau du serveur. En somme, toute la charge repose sur le serveur. Elle nécessite donc une puissance de traitement et une grande fiabilité de la part du serveur. Cela augmente le coût du système. Mais l'avantage est que l'architecture du système est simple.

b) Architecture client/serveur

Dans cette architecture, le client s'occupe des opérations frontales, tout ce qui est présentation, et le serveur du stockage des données, de la sécurité des données et du traitement des transactions. La charge du système est ainsi répartie entre le client et le serveur. Dans cette architecture, les données sont bien contrôlées. Par contre, le coût de développement (du client et du serveur) et de gestion est plus élevé. Le client a plus de charges que celui du système centralisé.

c) Mode SIG Web/internet

Ce mode est le même que le client/serveur. En fait, il combine la technologie d'Internet et du client / serveur. Il est simple et flexible, car le client est indépendant de la plate-forme et il nécessite seulement un navigateur Web. De plus, le protocole de communication utilisé est toujours HTTP. Par contre, l'architecture client/serveur (non orientée Web) doit avoir une plate-forme spécifique sauf dans certains cas où le langage de programmation utilisé est multi-plate-forme, cette

architecture peut supporter plusieurs protocoles de communication et elle nécessite toujours l'installation de l'application chez le client.

2.1.2.2 Composants de base de l'architecture SIG Web

Dans cette section, nous abordons les éléments constitutifs de l'architecture SIG Web de type client / serveur (voir Figure 11).

a) Client

Généralement, le client fait référence au navigateur Web de la machine de l'utilisateur. Dans le monde du SIG Web, c'est l'outil utilisé par les utilisateurs pour visualiser les données spatiales et exploiter les fonctions d'analyse et d'affichage [56]. Il existe deux types de clients. Le client léger constitue une interface pour communiquer avec le serveur et afficher le résultat [57, 51]). Le client lourd prend en charge le traitement des données et la communication avec la base de données. Il contient certaines fonctionnalités du SIG et soumet des requêtes au serveur pour demander des données géographiques. Il prend alors la forme d'extensions qui ajoutent des capacités au navigateur. Les différentes extensions sont Adobe Acrobat, Adobe Flash, Adobe SVG viewer, Apple QuickTime, Leaflet et OpenLayers.

b) Serveur

Un serveur SIG Web comporte généralement quatre composantes : un serveur Web, un serveur d'application, un serveur cartographique et un serveur de données.

- **Serveur Web** : Le serveur Web répond aux demandes envoyées par le navigateur Web via HTTP. Tout site Web publié sur internet doit avoir un serveur fonctionnant en arrière-plan. Apache est un exemple de serveur Web.
- **Serveur d'applications** : Ce serveur est un logiciel qui permet de développer, déployer et gérer un grand nombre d'applications dans un environnement distribué. Un serveur d'applications agit comme un intergiciel qui établit, maintient et met fin à la connexion entre le serveur Web et le serveur cartographique. Il gère également les requêtes simultanées et équilibre la charge entre les serveurs cartographiques. Du point de vue de concepteurs, l'objectif principal d'un

serveur d'application est de séparer la logique du métier de la logique de présentation et de celle de la base de données.

- **Serveur cartographique** : Ce serveur est le cerveau de toute l'application SIG Web. Il fournit des fonctionnalités spécifiques traditionnelles des SIG qui incluent [56] :
 - l'analyse spatiale (qui se rapporte à l'espace, à l'étendue) ;
 - le traitement de requêtes spatiales ;
 - le traitement de requêtes d'attributs ;
 - le géocodage ;
 - le géotraitement.

Ce serveur génère et fournit également des cartes dynamiques au client en fonction des requêtes des utilisateurs. Il peut générer des résultats sous forme :

- de données, comme la sélection de tous les ponts d'une ville avec leurs caractéristiques.
 - d'images graphiques, comme par exemple une requête « GetMap » demandant la carte de la ville de Sherbrooke.
- **Serveur de données** : Le serveur de données gère les données spatiales ou non spatiales dans un système de gestion de base de données relationnelle ou non relationnelle [56].

2.1.3 Etude de l'existant

2.1.3.1 Travaux sur les SIG Web

Il existe dans la littérature plusieurs SIG Web.

Plusieurs auteurs ont mis en œuvre un système SIG Web basé sur des serveurs de base de données tels Oracle et PostGIS, et sur des serveurs SIG comme GeoServer. Enfin OpenLayers est utilisée pour mettre en œuvre une variété de fonctions pour l'application cliente. C'est le cas de Zhang et Lan [46] et aussi le cas S. Kubota et al. [58] qui ont réalisé un projet traitant les sinistres tels que les inondations. Ces derniers ont également mis en place une application mobile pour faciliter la collecte des informations. Les techniciens se trouvant sur le site du sinistre peuvent ainsi utiliser leurs téléphones intelligents pour envoyer aux gestionnaires les informations sur les catastrophes munies de leurs localisations (coordonnées géographiques). Le système analyse automatiquement

ces informations et fait la mise à jour du système. Le gestionnaire va diffuser les informations concernant le désastre via le SIG Web [58]. Cette diffusion peut se faire en affichant les nouvelles informations sur la carte.

Dans une autre étude, Jiang et Zhang [45] ont aussi choisi une architecture SIG Web. Pour eux, il est important que leur service SIG soit accessible via le Web. Les composants sous-jacents du SIG doivent donc être compatibles avec le Web afin que le système devienne accessible sur Internet. Le SDK MapInfo MapXtreme pour Windows est choisi et ils le désignent comme l'outil pour le développement d'applications SIG Web le plus puissant et le moins coûteux. Les outils utilisés sont, Microsoft SQL Server pour la base de données, Microsoft IIS Web server comme serveur Web, MapInfo MapXtreme pour la cartographie et Active Server Pages (ASP) comme langage de programmation.

Zhang et al. [46] ont réalisé un système SIG Web pour une étude d'évaluation de performance de la technologie SIG Web. Leur objectif est de renforcer la puissance du serveur. Pour ce faire, les modifications qu'ils ont effectuées sont l'amélioration de la JVM Java afin de pouvoir utiliser la technologie servlet pour ainsi faire usage de java et de sa capacité à créer un système de caches (un cache pour les images, etc.) permettant de renvoyer le même résultat, mis en cache, à une éventuelle requête ultérieure. Enfin, le système est conçu, implémenté, testé et comparé à des modèles dans une série d'expériences.

2.1.3.2 Exemples des projets SIG pour la gestion des infrastructures routières

Il existe plusieurs projets faisant usage du SIG et du SIG Web pour la gestion des infrastructures routières.

Jiang et Zang [45] ont implémenté un SIG Web pour l'inspection de la qualité de construction des ponts et pour l'évaluation de ceux déjà construits. Le système prend en charge tout type de pont.

Kubota et al. [58] ont conçu un SIG Web pour partager des informations sur des catastrophes naturelles telles que des tremblements de terre, des inondations, des glissements de terrain, etc. Un prototype a été mis en application par l'administration locale, pour des cas de tempête de vent et d'inondations. La collecte des informations est faite via la soumission de courriels et de photos du site par des personnes se trouvant sur le lieu du sinistre.

Li et Xie [42] ont conçu un SIG Web pour la prise de décision sur la maintenance des autoroutes. La collecte des données sous forme d'images de la chaussée est faite par des véhicules de test et permet de détecter l'orniérage, les fissurations et autres détériorations visibles. Les indicateurs sur lesquelles les décisions se basent sont le nivellement de la chaussée, les dommages de la chaussée, la résistance de la chaussée et les tests antidérapants. De même, Wang et Yue [53] ont implanté un système SIG Web pour la gestion de projets reliés aux autoroutes.

Enfin Sandrone et Labiouse [43] ont mené une étude afin d'identifier et analyser les pathologies des tunnels en Suisse via un SIG. Les causes des pathologies identifiées à l'issue de l'étude sont liées à la construction, l'opération et l'environnement.

2.2 Optimisation des itinéraires et cartographie

Afin de réaliser le calcul d'itinéraire en cas de bris, il faut d'abord étudier les travaux de recherche théoriques ainsi que les projets pratiques réalisés comme les applications cartographiques de calcul d'itinéraires.

2.2.1 SIG et Réseau routier

Comme nous l'avons déjà dit, les SIG sont capables de représenter les réseaux routiers avec une approche vectorielle (elle représente les entités géographiques par des points, des lignes et des polygones) qui permet de les modéliser avec toutes les contraintes réelles, par exemple, représenter les zones scolaires traversées par celui-ci. Cette représentation a l'avantage de pouvoir se transposer en graphe. Une fois les réseaux routiers transposés en graphes, tous les algorithmes de recherche de chemins optimaux [3] [4], issus des années de recherche dans le domaine de l'informatique et de la théorie des graphes, peuvent être appliqués sur ceux-ci.

Toutefois, la plupart des algorithmes de recherche de chemins optimaux utilisent la vitesse moyenne de l'entité en déplacement et la distance à parcourir comme critères d'évaluation. Or, ces dernières ne reflètent pas la réalité sur le terrain étant donné la complexité du réseau routier dans la vie pratique. Il existe un nombre important d'éléments qui faussent cette vision très simpliste de la question, notamment les règles de signalisation (limitation de vitesse, interdiction de tourner à gauche ou à droite, feux, sens unique, etc.), le volume du trafic, l'état de la route, les zones scolaires, les zones à forte concentration de population, les zones de présence d'animaux, etc.

Face à ce constat, le SIG, avec sa capacité d'intégration de contraintes réelles, se révèle capable d'apporter des solutions. Il est capable d'intégrer, de gérer et d'analyser les données géographiques des réseaux routiers et leurs attributs [8].

Plusieurs travaux de recherche viennent appuyer l'idée que le SIG est un outil idéal pour représenter le réseau routier avec toutes ses contraintes.

Dunn et Newton [3] ont mis en évidence que la capacité à gérer et à relier différents types d'entités telles que les points (nœuds) et les lignes (arcs) est une autre force de l'approche des SIG pour la recherche des chemins. Cela s'exprime, par exemple, dans la capacité à incorporer des niveaux de difficulté comme la restriction de tourner à gauche ou à droite sur une route. Ces caractéristiques peuvent être attachées à des paires d'arcs en matière de temps de trajet. Selon l'angle de virage, la vitesse est réduite plus ou moins longtemps, par exemple, le virage sur un carrefour giratoire (en forme de cercle) fait souvent un angle de 180 degrés ou plus. Il nécessite donc plus de temps que le virage sur une intersection plane faisant souvent un angle de 90 degrés.

Dans la même étude, les auteurs précisent que l'évolution à long terme des SIG devra tenir compte d'une multitude de facteurs supplémentaires. Un grand nombre de ces variables a fait l'objet d'une longue tradition de recherche ; elles comprennent les files d'attente, la congestion, les heures de pointe, les goulots d'étranglement, les panneaux d'arrêt, les feux de circulation, les travaux routiers, les viaducs et tunnels, les rues à sens unique, la largeur de la route, le nombre de voies, les vitesses maximales et moyennes, et les virages à gauche ou à droite.

L'importance d'un SIG réside aussi dans sa capacité à combiner les bases de données traditionnelles et les cartes numériques afin de fournir une représentation graphique tangible des informations. Il offre également des options de modélisation spatiale qui mettent en évidence l'interdépendance entre divers facteurs, par exemple, la distance entre les points de départ et de destination dépend de types de routes (il n'y a pas des feux de circulation sur une autoroute) [12].

Pour illustrer davantage la capacité d'un SIG à représenter le réseau routier, Waters [5] indique que les applications géographiques dans le domaine du transport sont devenues significativement populaires ces dernières années, si bien qu'elles sont maintenant couramment désignées par l'acronyme SIG-T. Il ajoute aussi que les réseaux routiers étaient parmi les premières structures à

être représentées dans le SIG et que l'analyse de réseau demeure l'une des recherches les plus importantes et persistantes dans la science d'information géographique.

Il faut souligner que la pratique de la géomatique est encadrée par des principes rigoureux et des normes précises, comme la norme définie par Open Geospatial Consortium (OGC) et le principe de l'Architecture orientée service (AOS), qui définissent l'interopérabilité des données et des systèmes. Par exemple, au ministère des Transports du Québec (MTQ), la division de la géomatique (DGéo) est en grande partie garante de la mise en application de ces meilleures pratiques [1].

De nombreux développements conceptuels ont également contribué à l'émergence des SIG-T. Ces concepteurs incluent les travaux dans la recherche et la programmation des opérations qui ont conduit à de nouveaux algorithmes pour l'analyse du plus court chemin, les procédures de routage, la résolution du « problème de transport » de la programmation linéaire et la segmentation dynamique des liens au sein du SIG-T [5].

2.2.2 Problème de recherche du meilleur chemin

La recherche de chemin est un problème commun à plusieurs disciplines, telles que la recherche opérationnelle, les réseaux informatiques et l'intelligence artificielle. Généralement, un chemin optimal pourrait avoir plusieurs significations selon les contraintes imposées. Il peut désigner le chemin le plus rapide, le facteur considéré dans ce cas étant le temps, ou le chemin le plus court, le facteur considéré étant la distance [2]. Notre étude s'intéresse au meilleur chemin en termes de temps ; il s'agit là donc du chemin le plus rapide.

2.2.2.1 Classification de problème de recherche de meilleur chemin

La plupart des chercheurs ont regroupé les différents types de problèmes de recherche du meilleur chemin selon les catégories suivantes [10], [11] :

a) Paire unique (point-to-point)

Cette catégorie d'algorithmes calcule le plus court chemin d'un nœud source $s \in V$ à un nœud de destination $t \in V$. Les algorithmes Dijkstra, Bellman-Ford et A* résolvent les recherches de cette catégorie d'algorithmes [75].

b) Origine unique (single-source)

Le principe de cette catégorie d'algorithmes est tel qu'on choisit un sommet fixe $s \in V$ appelé source ou origine, et on calcule le plus court chemin vers tous les sommets $v \in V$. Les algorithmes Dijkstra, Bellman-Ford et A* font partie de cette catégorie d'algorithmes [75].

c) Plusieurs-à-plusieurs (Many-to-Many)

Cette catégorie d'algorithmes prend un ensemble de sommets sources noté $S = \{s_i: 0 \leq i < |S|\} \subset V$ et un ensemble des sommets destinations noté $T = \{t_j: 0 \leq j < |T|\}$. Elle calcule ensuite pour tous les (i,j) les distances $d_{ij} = d(s_i, t_j)$ telles que d_{ij} est le plus court chemin de s_i à t_j . L'algorithme Floyd-Warshall et l'algorithme de Johnson appartiennent à cette classe [76].

d) Recherche pour tous les couples de sommets (all-pairs)

Comme la recherche « plusieurs-à-plusieurs », cette catégorie d'algorithmes permet de calculer le plus court chemin pour chaque couple (S, T) , mais avec la particularité que les ensembles S et T comprennent tous les sommets du graphe ($S = T = V$). Elle constitue donc un cas spécial de la classe « plusieurs-à-plusieurs ».

2.2.3 Algorithme de routage

La représentation du réseau routier sous forme d'un graphe est toujours nécessaire avant de penser à des algorithmes de routage. L'une des représentations d'un graphe correspond à une configuration des segments de route en arcs, les intersections sont modélisées par des nœuds et toutes les mesures importantes relatives aux segments des routes (à considérer par le projet) par les poids des arcs. Une fois ce travail fait, il faut choisir un algorithme.

Il existe plusieurs algorithmes qui calculent le plus court chemin relatif à des domaines spécifiques allant de la recherche opérationnelle (théorie des graphes) à l'informatique. L'un des algorithmes les plus performants qui répond au problème classique du plus court chemin est celui de Dijkstra issu de la théorie des graphes et qui est déjà très utilisé dans beaucoup de domaines. On peut citer, par exemple, son utilisation dans le réseau informatique avec une implémentation pratique (réelle) qui est celle du protocole OSPF sur les routeurs CISCO. Il est aussi utilisé dans les principaux logiciels de cartographie et de SIG dont ArcInfo, GFIS et AUTOROUTE [3].

Un problème important du plus court chemin est qu'il crée une surcharge sur un axe donné, en délivrant tout le trafic sur cet axe. L'algorithme de Fulkerson et de Ford [59] vient résoudre ce problème de surcharge d'un axe introduit par l'algorithme de Dijkstra. Cet algorithme choisit par exemple les trois meilleurs chemins et distribue le flux entre les trois. CISCO a implémenté cet algorithme dans son protocole de routage EIGRP sur les routeurs CISCO récents. Dunn et Newton [3] ont montré qu'il était possible d'utiliser l'algorithme de Ford et Fulkerson [59], pour des cas tels que l'évacuation des victimes, lors d'incidents, vers plusieurs centres de réception, par exemple les hôpitaux, en recherchant des routes optimales et en distribuant le flux entre ces dernières.

2.2.4 Astuces d'accélération

Le problème du plus court chemin dans un graphe a été largement résolu en 1959 par Dijkstra. Ce dernier a réalisé un algorithme résolvant le problème de « *origine unique* » pour un graphe $G = (V, E)$ avec n sommets et m arcs.

Or, l'application de l'algorithme de Dijkstra ainsi que ceux de Fulkerson et Ford sur un réseau routier très large sont lents [10,11]. Il existe un bon nombre de techniques d'accélération qui misent sur le prétraitement du graphe en générant des données intermédiaires qui servent à accélérer les requêtes pour obtenir le plus court chemin.

Dans la littérature, il existe un certain nombre de techniques de prétraitement telles que « Reach » [19], « Arc flags » [24], « A* Search », « Precomputed Cluster Distances » [11], « Compressed Path Databases » [77], « Cantration Hierarchies » [28, 29], etc.

Schultes [11] préconise, afin d'accélérer ce genre de traitement, l'utilisation de techniques qui répondent aux spécifications suivantes :

- que les réponses aux requêtes soient les plus rapides possible,
- que les résultats soient précis,
- que les méthodes puissent s'adapter à l'évolution du réseau,
- si l'approche utilise des prétraitements, elle doit être suffisamment rapide pour supporter des réseaux routiers de grandes tailles
- que les données intermédiaires de prétraitement occupent des espaces modérés,
- que les mises à jour des poids des segments de routes ou le remplacement des fonctions de coût soient possibles.

Le volume des données du réseau routier étant considérable, il devient difficile de calculer un itinéraire rapidement. C'est pourquoi de façon générale des techniques de prétraitement sont utilisées. Elles consistent à calculer tous les itinéraires possibles du réseau routier à l'avance. Ces itinéraires sont stockés et utilisables au besoin. Cela a un coût en termes de mémoire qu'il est difficile d'éviter. Un compromis entre le coût du prétraitement et l'espace mémoire devient une nécessité. Malheureusement, le prétraitement est mal adapté au réseau routier, car ce dernier est sujet à des changements constants (puisque le trafic routier est un évènement qui se passe en temps réel, et que chaque seconde est importante). C'est ainsi qu'on s'oriente vers d'autres algorithmes relevant ce défi.

2.2.4.1 Techniques hiérarchiques

Les techniques hiérarchiques sont des méthodes qui ont pour but d'exploiter la hiérarchie des réseaux routiers, c'est-à-dire la subdivision du réseau routier en sous réseaux (niveaux d'importance des routes). Bien que ces techniques soient actuellement très populaires, elles ne fournissent pas toujours les solutions optimales. Ces techniques, comprenant Reach et Cantration Hierarchies (CH), ont été utilisées par Gutman [19] sur un graphe de 400 000 sommets et le traitement s'avère 10 fois plus rapide que celui fait par l'algorithme de Dijkstra.

Un exemple pratique de techniques hiérarchiques (Cantration Hierarchies) est l'Open Source Routing Machine (OSRM) qui a été testé par UBER sur des données comportant toutes les routes du monde. Ce test a pris 12 heures pour construire le graphe contracté. Bien que les requêtes soient très rapides (moins de 100 millisecondes) après la phase de prétraitement, les concepteurs conviennent que cela est malheureusement difficile à appliquer au trafic routier étant donné que l'apparition d'un changement sur le réseau routier implique un nouveau prétraitement sur la partie affectée. Cela est inadapté au trafic routier qui nécessite des réponses en temps réel.

2.2.4.2 Techniques orientées vers le but

Contrairement à l'algorithme de Dijkstra qui analyse tous les sommets d'un graphe, les techniques dirigées vers le but (« **Goal-Directed Techniques** ») orientent la recherche vers la cible. Elles ne prennent en considération que les sommets situés dans la direction de la destination. Elles exploitent soit l'intégration (géométrique) du réseau ou les propriétés de graphes, telles que l'arbre

de plus court chemin. Cette catégorie contient, entre autres, les algorithmes Arc flags, A* Search, Precomputed Cluster Distances et Compressed Path Databases [11, 77].

2.2.4.3 Techniques Bounded-Hop

Les techniques « Bounded-Hop » préconisent le calcul à l'avance des distances entre les paires de sommets afin d'ajouter des raccourcis virtuels au graphe. Ainsi, elles utilisent seulement les distances déjà calculées plutôt que le graphe en entrée. Les algorithmes basés sur ces techniques sont Labeling Algorithm, Pruned Highway Labeling et Transit Node Routing [11, 12].

2.2.4.4 Combinaison des techniques

D'après la littérature, les combinaisons de ces techniques donnent souvent des bons résultats. Parmi les travaux menés en ce sens, on peut citer la combinaison de Transit Node Routing avec Arc Flags [24]. Aussi, plusieurs travaux combinent les méthodes hiérarchiques comme CH ou Reach avec celles de goal-directed comme ALT ou Arc Flags [25] [26]. Il faut noter que l'usage de Reach-based routing avec l'algorithme ALT donne un meilleur résultat que celui des algorithmes Reach-based routing et ALT séparément [25].

2.2.4.5 Améliorations de techniques existantes

Plusieurs améliorations peuvent être appliquées aux techniques précédentes. Les algorithmes résultants sont considérés comme les mieux adaptés quand il est question de traiter des problèmes liés à des réseaux dynamiques ou de traiter avec des fonctions objectives multiples, c'est-à-dire que le coût peut être défini à l'aide de diverses métriques. Par exemple, nous pouvons imaginer l'utilisation du temps pour certains tronçons et du péage pour d'autres tronçons. Un exemple d'algorithme amélioré est PHAST [27], construit au-dessus de Contraction Hierarchies. Des travaux avec des fonctions objectives multiples se trouvent dans des études concernant les Contraction Hierarchies [29, 28].

La majorité de ces améliorations optimise de façon significative le temps de réponse aux requêtes, tout en faisant un bon compromis sur l'utilisation de mémoire. Toutefois, elles sont restées inefficaces face aux exigences du trafic routier qui est sujet à des changements constants. Ainsi, pour faire les mises à jour, il faut un prétraitement. Par contre, des algorithmes arrivent à répondre correctement aux exigences du trafic routier via des acrobaties. Parmi ces algorithmes on a A*, de

la catégorie des techniques dirigées vers le but, qui répond très bien à toutes ces exigences avec facilité. A* devient un peu lent dans ses réponses quand la taille du réseau routier monte à l'échelle d'un continent.

2.2.5 Exemple d'études sur le calcul d'itinéraires

Cette section montre des études menées pour des applications non commerciales, notamment des études sur les produits alimentaires périssables et des études sur le transport de victimes d'accidents vers les centres des urgences.

Cependant, il est important de noter qu'aucun de ces travaux n'a utilisé des données spatiales à part celui d'Abousaeidi et al. [12].

Un des domaines pour lequel beaucoup de recherches ont été menées est celui du transport des aliments. En effet, les produits frais sont très sensibles à la chaleur. L'avalissement de ces produits est accéléré par cette dernière quand le temps passé sur la route devient trop long. De plus, il est difficile de créer toutes les conditions nécessaires afin de bien conserver les caractéristiques nutritionnelles de ces produits frais lors de la livraison. La question de la distribution des produits frais à travers le réseau routier est donc d'un intérêt particulier dans le domaine de l'industrie agroalimentaire.

Osvald et Stirn [13] ont porté une attention particulière aux fluctuations du temps de déplacement dans le domaine de la distribution, surtout que le temps requis est un facteur critique pour les aliments périssables. Ces fluctuations peuvent prolonger le temps que les aliments passent sur les véhicules. Plusieurs auteurs [60, 61, 62] ont mené des travaux qui traitent des problèmes de routage avec un temps de trajet stochastique (on peut changer d'itinéraire au cours du trajet, par exemple, à cause du trafic).

Une étude menée par Abousaeidi et al. [12] fait usage de la modélisation SIG pour déterminer le meilleur chemin pour la livraison des légumes frais à Kuala Lumpur (Malaisie). À l'issue de leur étude par régression, ils ont montré que les facteurs qui influencent le plus la durée du trajet sont le trafic routier, la distance à parcourir, la densité de la population, la présence de zones scolaires, les restrictions de voies et les zones d'habitation.

Il existe aussi des études qui ont été effectuées dans le domaine de la distribution du lait par Tarantilis et Kiranoudis [15] et de la viande fraîche par J.M. Belenguer et al. [16], ainsi que Tarantilis et Kiranoudis [14].

D'autre part, dans le même registre de calcul d'itinéraire, mais dans un autre domaine d'application, Kenyon et Morton [62] ont mené une étude pour identifier les routes les plus courtes pour accéder aux urgences à partir du lieu d'accident. Pour arriver à un bon résultat, ils considèrent les paramètres tels que la restriction de route à sens unique, la distance à parcourir, les limites de vitesse, etc.

2.2.6 Applications cartographiques de calcul d'itinéraire

Dans cette partie nous abordons les applications cartographiques de calcul d'itinéraires. Nous nous intéressons aussi aux données qu'elles utilisent pour calculer les itinéraires. D'abord, nous exposons les applications telles que Google Maps, Apple Plans et Bing Maps qui sont conçues traditionnellement pour être consultées sur des ordinateurs. Ensuite, nous abordons les applications embarquées sur terminaux GPS. Ces dernières sont, soit déployées sur le tableau de bord d'un véhicule, soit sur un autre appareil de navigation. Elles nécessitent donc l'utilisation d'un *Système mondial de positionnement* (GPS) pour localiser les véhicules roulants (ou tout autre appareil de navigation en état de déplacement) en se basant sur des signaux captés des satellites pour les systèmes GPS. Les GPS sont à l'origine de l'expansion de l'utilisation d'applications de géopositionnement. Puis avec l'avènement des téléphones intelligents (GPS intégré), ces applications sont devenues populaires et très faciles d'accès. Parmi ces applications, il y a celles qui proposent d'alléger la facture de péage en vous incitant à faire des sorties et entrées à l'autoroute pour contourner les points de péage ou amoindrir le prix de l'essence en listant les prix du carburant aux stations environnantes. Des exemples de ces applications sont *Gasoil Now*, *Essence Free* et *AutorouteEcho*. Elles nécessitent des appareils équipés de GPS tels que ceux de *TomTom*, *Coyote NAV*, *Garmin*, *CoPilot*, *Magellan* et *Sony*. Nous abordons ici seulement les systèmes *TomTom* et *Garmin*.

2.2.6.1 Google Maps

Présente sur toutes les plateformes, Google Maps est l'une des applications les plus populaires dans le domaine cartographique. Elle constitue aussi une base à plusieurs autres applications tierces

grâce à son API. Google produit ses propres données et il couvre le monde entier par ses milliers de cartes. Il propose plusieurs services, dont le calcul d'itinéraire qui offre plusieurs modes : voitures, transport en commun, vélo et marche. On peut même choisir d'éviter ou non les autoroutes et les postes de péage ou bien modifier le parcours par le curseur sur la carte. Google Maps était la première à fournir une alternative pour le transport en commun [30, 31]. Un exemple illustratif du calcul d'itinéraire de Google Maps est fourni à la Figure 7.

Il est très difficile de connaître les données utilisées par Google Maps pour calculer l'estimation du temps d'arrivée (ETA). Cependant d'après Richard Russell, un ancien concepteur de Google interrogé en juillet 2013 dans le journal Quora [20], Google maps calcule l'estimation du temps d'arrivée (ETA) selon la disponibilité de données dans une zone. Ces données peuvent être la vitesse, soit indiquée ou recommandée (selon les types des routes), les données historiques des usagers, comme les vitesses moyennes à des périodes particulières de la journée et la durée réelle du trajet, le trafic en temps réel, etc.

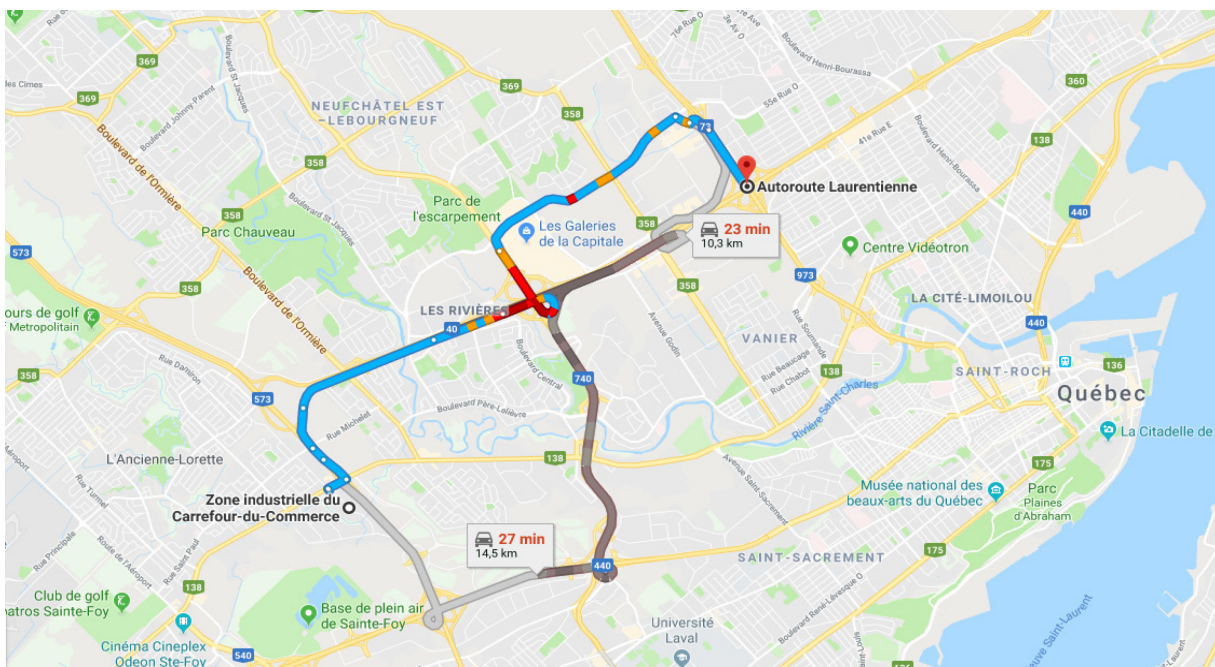


Figure 7 - Exemple de calcul d'itinéraire de Google Maps [30]

2.2.6.2 Apple Plans

Cette application, développée par Apple pour iOS et macOS, permet la navigation routière. Elle a utilisé la cartographie de Google Maps pendant cinq ans. Depuis juin 2012, Apple plans utilise son

propre système de cartographie, mais utilise des données provenant d'autres fournisseurs dont le principal est le fabricant néerlandais de TomTom. Dans son calcul d'itinéraires, elle considère l'heure, le trafic, la position et l'emploi du temps en usant du calendrier où sont marqués les rendez-vous. Elle offre une option de recherche pour trouver des endroits tels que station-service, restaurant, etc. Elle prend en considération le transport en commun également grâce à sa fonction « Transport ». On peut aussi planifier l'itinéraire par la voix via siri [32, 33].

2.2.6.3 ViaMichelin [34]

L'application Via Michelin couvre 45 pays en Europe ; elle utilise ses propres données en plus des celles de la société néerlandaise produisant les cartes Tele Atlas. Dans son processus de calcul d'itinéraires, elle offre le trajet le plus rapide, le trajet le plus court, le trajet recommandé, le trajet de découverte et le trajet économique. Elle prend également en considération, dans son calcul, des paramètres comme le type de voiture et la consommation du carburant. Elle offre aussi le choix d'éviter les autoroutes et/ou les postes de péage.

2.2.6.4 Waze

Waze est une application de navigation GPS participative. Les usagers se partagent les informations sur le trafic. Par exemple, un utilisateur pris dans un embouteillage peut le signaler, de même s'il est en face d'un accident. Cela permet aux autres usagers d'éviter les axes où il y a un incident. Elle offre aussi des choix tels que, l'évitement des routes à péage ou l'affichage 3D. Elle intègre également, dans son calcul de trajet, la consommation de carburant. L'interface Waze étant inspirée des GPS pour les automobiles, elle est donc plus pratique pour les conducteurs que les applications comme Maps ou Plans. Google a racheté Waze récemment [35, 36].

Pour calculer le meilleur chemin, Waze se base sur les historiques de vitesses, les feux de circulation, les ronds-points, les limites vitesses, les embouteillages en temps réels, les détails sur l'endroit où les conditions de circulation sont défavorables, les horaires journaliers, les jours de la semaine et les fins de semaine [23].

2.2.6.5 Mappy

Mappy est une application de calcul d'itinéraire présente en Europe ; elle calcule l'itinéraire en comparant tous les modes des déplacements (transport en commun, voiture, train, vélo, marche ...)

à l'image de Maps et Plans. Elle propose des choix de calcul d'itinéraires selon le trajet le plus rapide ou le plus court (voir Figure 8). Elle offre donc moins d'options que ViaMichelin. Dans son calcul, elle inclut des options telles qu'un itinéraire avec ou sans postes de péage, le type de véhicule et le type de carburant. Elle avise également l'utilisateur du retard qui est occasionné par le trafic, par rapport au calcul de départ. L'information sur le trafic est actualisée toutes les deux minutes.

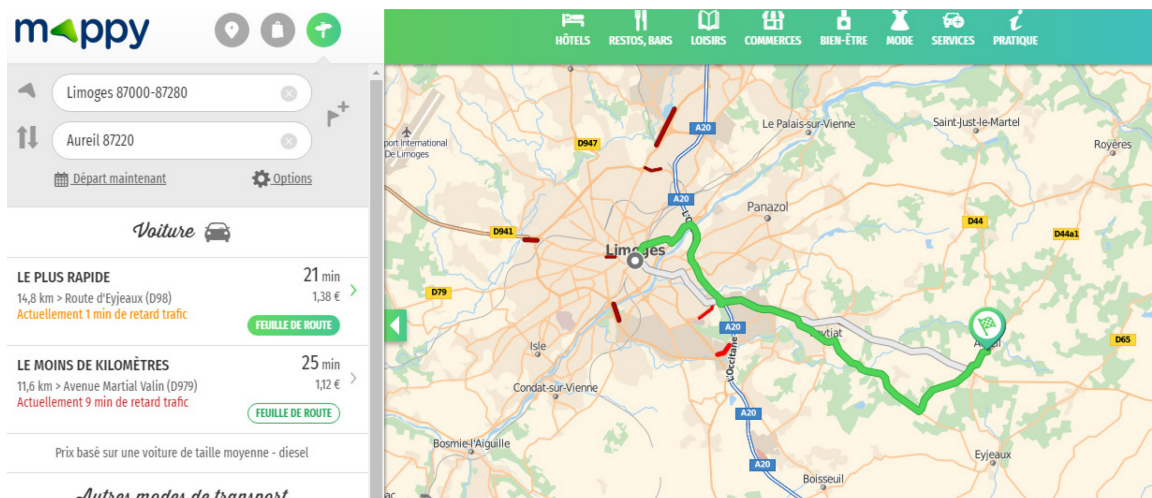


Figure 8 – Calcul d'itinéraire par Mappy (source : <https://fr.mappy.com>)

2.2.6.6 Sygic

C'est une application créée en Slovaquie qui utilise la carte TomTom. La particularité de Sygic est qu'elle offre une navigation hors ligne en étant téléchargée sur son appareil personnel. Elle est disponible dans plus de 200 pays avec 30 langues différentes [37]. Elle propose des options comme la navigation 3D, les instructions vocales pas-à-pas, l'affichage des informations du trafic en temps réel et les choix d'itinéraires tels que le plus rapide, le plus court, le plus économique, etc. Elle permet aussi d'éviter les postes de péage et les routes non bitumées,

2.2.6.7 Bing Maps

Bing Maps est une application qui utilise généralement les données de Navteq pour construire ses cartes et calculer ses itinéraires, mais quelquefois celles de OpenStreetMap et Ordnance Survey qui est une agence exécutive du gouvernement de Royaume-Uni chargé de la cartographie. Elle

calcule des itinéraires à pied et en voiture, et elle permet aussi d'éviter les péages et les autoroutes comme illustré à la Figure 9. De plus, elle inclut depuis 2010 les transits (correspondances) en bus, métro et train. Par contre, ces dernières ne sont disponibles jusqu'à présent que dans 11 villes américaines. Elle montre également l'information du trafic en temps réel sur la plupart des routes via le service ClearFlow, ce qui permet aux conducteurs d'éviter les congestions. ClearFlow montre les routes alternatives avec les conditions du trafic et prend également en compte les événements sportifs, les heures de la journée et les conditions climatiques [38].

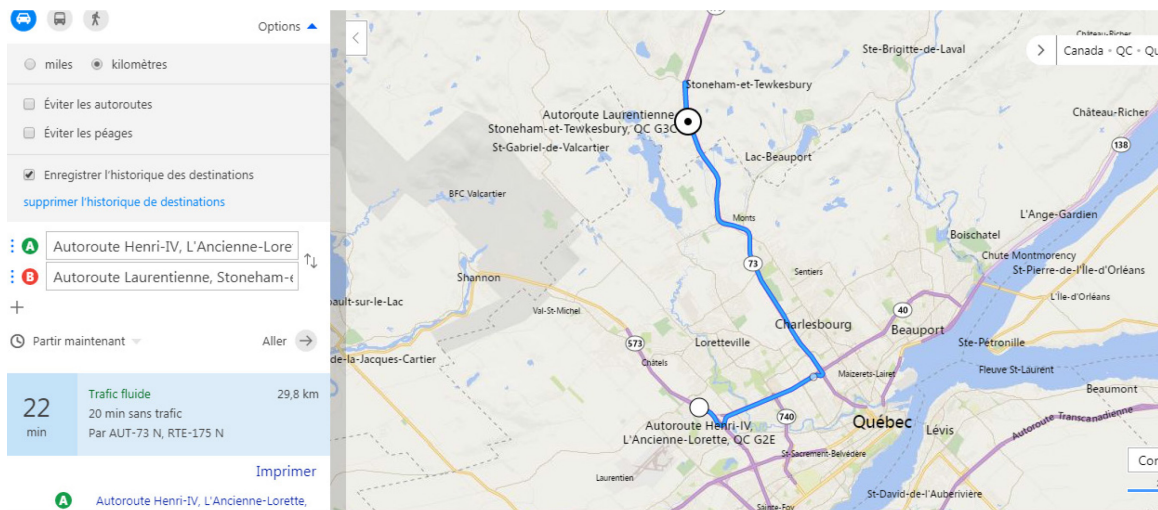


Figure 9 – Calcul d'itinéraire par Bing Maps [38]

2.2.6.8 Uber

La compagnie UBER est une entreprise technologique qui développe et exploite des applications mobiles permettant aux clients d'avoir accès aux taxis les plus proches.

Nous ne disposons pas beaucoup d'information sur les données servant au calcul de l'ETA d'Uber, mais nous savons que celle-ci utilise de données telles que les distances, les limites de vitesse, les routes à sens unique, les restrictions de virage, etc. [21]

2.2.6.9 TomTom

La compagnie TomTom a été créée par deux étudiants néerlandais ; leur société est devenue chef de file dans le marché de GPS mobiles. TomTom fabrique des GPS mobiles ou embarqués et développe des applications dédiées au calcul d'itinéraires [39].

TomTom utilise également des données similaires à celles de Waze, pour calculer des itinéraires [22].

2.2.6.10 Garmin

La compagnie Garmin a vu le jour grâce à deux ingénieurs américains. Elle s'est imposée dans le marché des GPS via ses produits de navigation pour automobile StreetPilot, Nüvi et Drive. Mis à part le GPS pour automobile, elle fabrique également des équipements pour l'aviation (appelé avionique) et les motocyclettes (appelé Zūmo). Elle développe aussi des applications GPS pour iPhone, Android et Windows Phone [40].

2.2.6.11 Conclusion

Dans la vie quotidienne, on peut bien facilement estimer le temps de parcours d'un trajet donné. Cette estimation est beaucoup plus précise quand il s'agit d'un trajet que l'on effectue souvent. Cela signifie effectivement que le cerveau se base non seulement sur un certain nombre de critères sur le trajet, mais aussi sur l'historique acquis par l'expérience. Il semble qu'au niveau technique beaucoup d'applications utilisent également le même principe pour calculer le plus court chemin. C'est dans cette logique qu'Alazab et al. [4] ont mené une étude dans laquelle ils déterminent l'heure d'arrivée en fonction de plusieurs facteurs tels que la distance, les travaux routiers, la congestion du trafic et la vitesse de conduite. Dans l'avenir, ce type d'information sera de plus en plus intégré dans les applications.

Chapitre 3 Conception et réalisation

La conception d'un SIG pour les infrastructures routières est une entreprise complexe. Le système que nous avons conçu est donc un prototype limité. La limite du prototype que nous avons choisi est la ville de Québec. La ville de Québec qui compte six arrondissements (voir Figure 10) est la capitale de la province du Québec qui est située dans la région administrative de la Capitale-Nationale. Ce choix est motivé par le fait que la ville de Québec possède des infrastructures riches et variées comportant presque tous les types d'infrastructures routières existants dans la province. La conception d'un SIG ou même d'un prototype limité présente tout de même un défi.



Figure 10 – Ville de Québec

Pour implémenter notre système, nous avons choisi une approche du type SIGWeb (Système d'information géographique Web) qui, selon nous, est l'outil idéal pour le développement de systèmes de gestion des infrastructures routières, car il permet d'intégrer, en plus de données habituelles, les composantes géographiques et d'y accéder via le Web.

Un tel système doit d'abord être nourri de toutes les informations pertinentes sur la ville de Québec. Notre principale source de données, qui est aussi la base de notre modèle conceptuel, est le Ministère des Transports du Québec (MTQ). Toutefois les données du MTQ sont incomplètes et

ne nous permettent pas de réaliser le système comme il doit l'être. En particulier elles ne contiennent pas d'information sur les limites de vitesse et sur certains ponts. Nous avons donc dû utiliser d'autres sources de données pour combler ces manques. Par exemple, nous avons utilisé les données d'OpenStreetMaps (OSM) pour éliminer le problème d'absence de données sur les limites de vitesse.

3.1 Méthodologie

La méthodologie suivie pour la réalisation du projet dans ce mémoire est décrite en quatre phases :

- Analyse des besoins ;
- Analyse et conception du système ;
- Analyse des informations ;
- Implémentation.

Cette approche est dérivée de celle proposée par Elmasri et Navathe [70].

3.2 Analyse des besoins

Les grandes lignes de l'analyse des besoins sont, entre autres, la compréhension du domaine et du problème par le biais de la collecte d'informations, de l'étude des systèmes existants et les spécifications des utilisateurs du système. D'après Béné [66], au-delà des aspects techniques, il est d'abord impératif de planifier et analyser les besoins du monde réel. C'est pendant cette étape que les spécifications fonctionnelles du système à développer sont exprimées. Pour réaliser cette étape, il a été nécessaire de collaborer avec les initiateurs du projet en génie civil, à travers des rencontres et échanges afin de bien identifier les besoins des futurs usagers du système. L'une des exigences fixées était de suivre la norme de description pour le réseau routier établie par ministère des transports du Québec (MTQ). En plus de cette collaboration, nous nous sommes appuyés sur la littérature pour mieux identifier ces besoins.

Entre autres le système doit :

- représenter et géolocaliser les infrastructures routières (routes et ponts) sous forme cartographique ;
- publier les informations relatives aux infrastructures routières ;
- présenter une carte de fond pour une meilleure lisibilité ;

- fournir un service de calcul d'itinéraire ;
- assurer la continuité du service routier même en cas d'incident ;
- être accessible en tout temps et de n'importe où ;
- avoir une interface conviviale et facile d'utilisation.

3.3 Analyse et conception du système

La conception est l'étape définissant le plan à suivre afin de respecter les exigences du système jusqu'à sa réalisation.

Notre système, dont l'architecture est décrite à la Figure 11, étant orienté Web, comprend quatre composantes : un client, un serveur cartographique, un serveur de base de données et un module de calcul d'itinéraire. Dans notre architecture le serveur Web est intégré au serveur cartographique.

Le client et le serveur interagissent conformément à la norme de l'OGC (Open Geospatial Consortium).

Le client envoie des requêtes au serveur cartographique demandant soit des informations sur les routes et/ou les ponts, soit le calcul d'itinéraire.

Le serveur cartographique reçoit les requêtes du client, il cherche les informations stockées dans la base de données qui sont nécessaires au traitement de requêtes à partir desquelles il fait une demande au module de calcul d'itinéraire selon le type de la requête du client. Le résultat est envoyé au client.

Le serveur de base de données est responsable de la sauvegarde de toutes les données (géographiques ou non) nécessaires à ce travail.

Le module de calcul d'itinéraire, qui est intégré à la base de données, reçoit une requête du serveur cartographique munie de coordonnées comprenant deux points et lui retourne l'itinéraire entre les deux points.

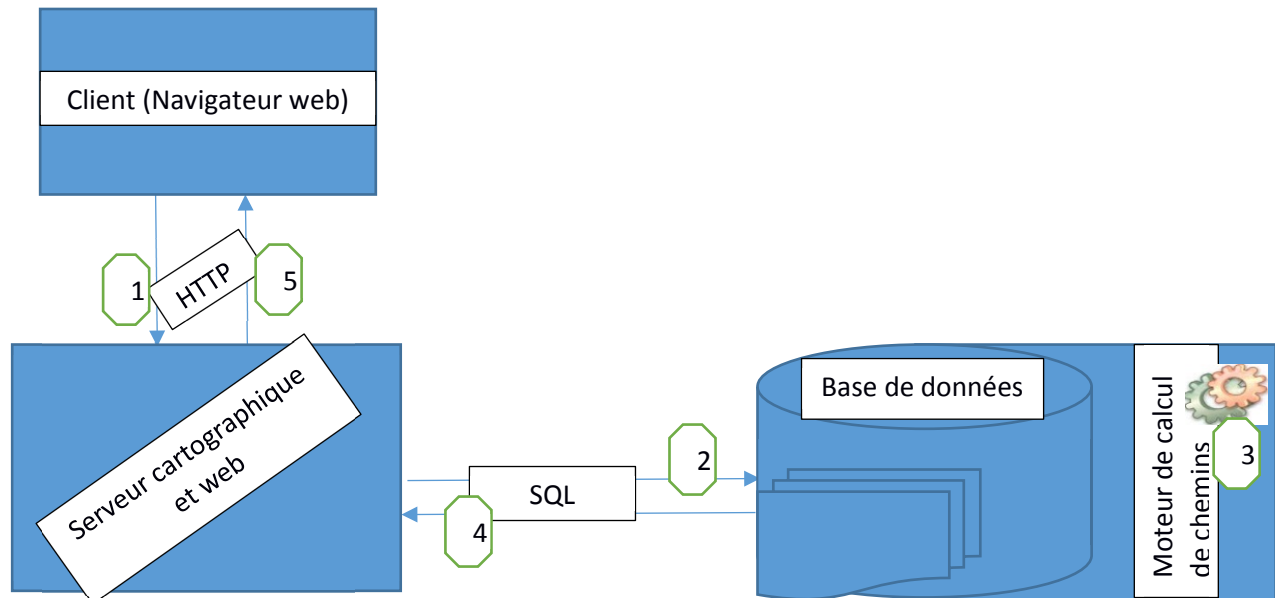


Figure 11 - Architecture du système

3.4 Analyse des informations

L'analyse est l'étape consistant à recueillir les informations nécessaires à la mise en place de la base de données. Dans notre système, nous avons besoin des informations sur les routes et les ponts de la ville de Québec. Ces informations concernent à la fois les attributs, la localisation et les aspects géométriques. Comme ce projet est destiné au ministère des Transports du Québec (MTQ), notre modèle conceptuel doit suivre les normes établies par le MTQ. Nous avons donc consulté la documentation sur les normes du MTQ [71] pour répondre à cette exigence de conformité. Nous devons, à partir de cette information, établir un modèle conceptuel, un modèle logique et enfin effectuer une analyse des données.

3.4.1 Modèle conceptuel

Le modèle conceptuel de données vise à réaliser une structure logique et physique pour la base de données afin de répondre aux besoins exprimés par les utilisateurs du système.

Notre modèle conceptuel suit la description de la MTQ. Selon le MTQ, le réseau routier se divise en six catégories de route comme indiqué sur la figure 12. Ces catégories sont les autoroutes, les routes nationales, les routes régionales, les routes collectrices, les artères (elles n'existent pas dans nos données réelles) et les routes locales. Ces catégories sont établies selon les limites de vitesse,

la démographie des agglomérations reliées par la route, etc. Chaque catégorie est ensuite subdivisée en tronçon et ce dernier en sections. Finalement, une section est constituée de sous sections. Enfin, tous les ouvrages d'art (ponts, ponceaux, tunnels...) sont situés quelque part sur les sections selon cette subdivision du réseau routier. D'après le MTQ [72] :

- une route est une voie carrossable du domaine public reliant entre eux des pôles (centres d'intérêt) économiques ou politiques. Par exemple, la route 175 (boulevard Talbot) reliant la ville de Québec à la ville de Saguenay est une route ;
- un tronçon est une partie de route délimitant une portion régionale du réseau routier. Par exemple, la partie de la route 175 située entre les bretelles d'accès aux routes 167 et 169 est un tronçon ;
- une section est une partie de route établie selon des critères de gestion administratifs et opérationnels du réseau routier. Par exemple, considérant la route 175 dans la ville de Québec, le début du pont de Québec (au sud) jusqu'à la fin du Pont (au nord) forme une section ;
- une sous-section (sous-route) est une chaussée distincte de chaînage continu à l'intérieur d'une section. Par exemple, considérant le même pont de Québec portant la route 175, sa chaussée de droite est une sous-section et sa chaussée de gauche pour la circulation inverse est une autre sous-section.

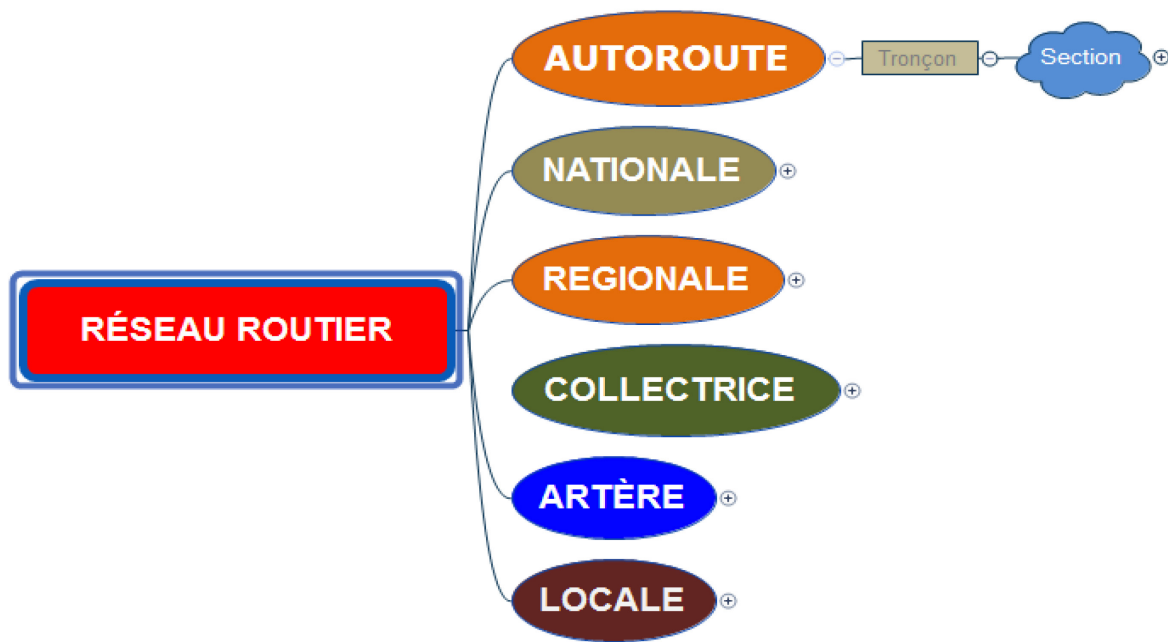


Figure 12 - Modèle conceptuel de données

À partir d'une section, qui est la portion de route où les ouvrages d'art (Pont, tunnel, viaduc, etc.) sont situés, il est possible d'étendre notre modèle conceptuel en sélectionnant le bouton plus (+ visible sur la Figure 12) de la partie « section » pour découvrir les différents types d'ouvrages d'art. Comme notre intérêt dans ce projet concerne les ponts, les Figures 13, 14 et 15 montrent les différents types des ponts avec leurs caractéristiques représentés par notre modèle conceptuel. Par exemple, sur une section, il est possible d'avoir des ponts à dalle épaisse, des ponts à poutres caissons, des ponts à poutres triangulés et des ponts à âme pleine. Nous obtenons des informations détaillées sur les infrastructures représentées dans le modèle via la sélection des boutons plus (+) de la figure. Les Figures 14 et 15, qui fournissent des informations relatives aux ponts à poutres triangulés et aux ponts à âme pleine en sont des exemples.

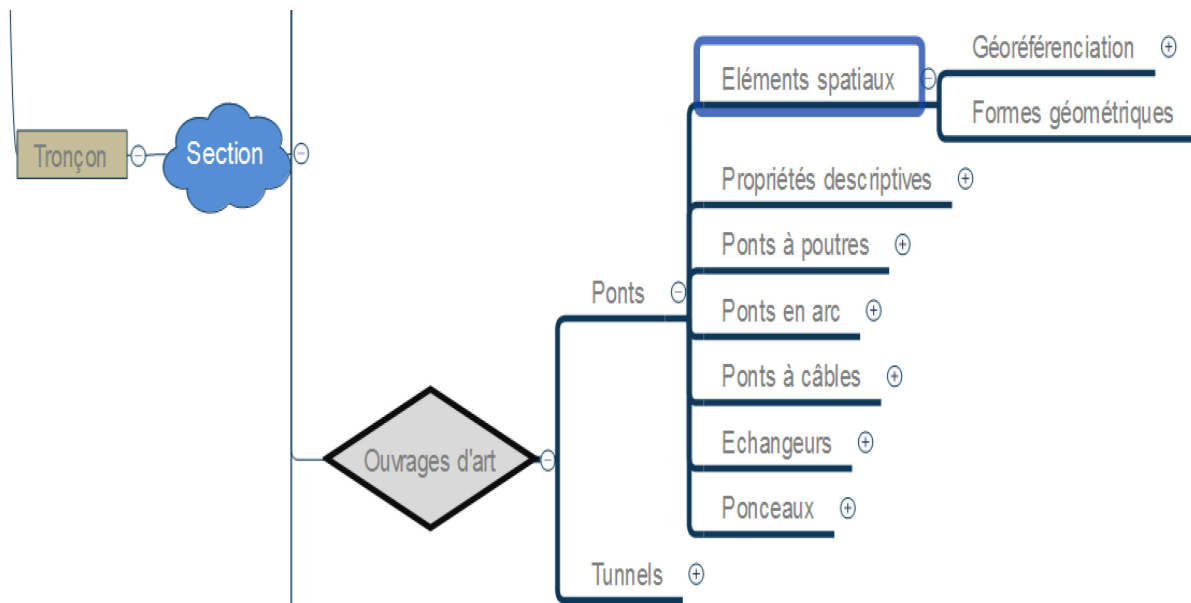


Figure 13 - Modèle conceptuel de données (suite 1)

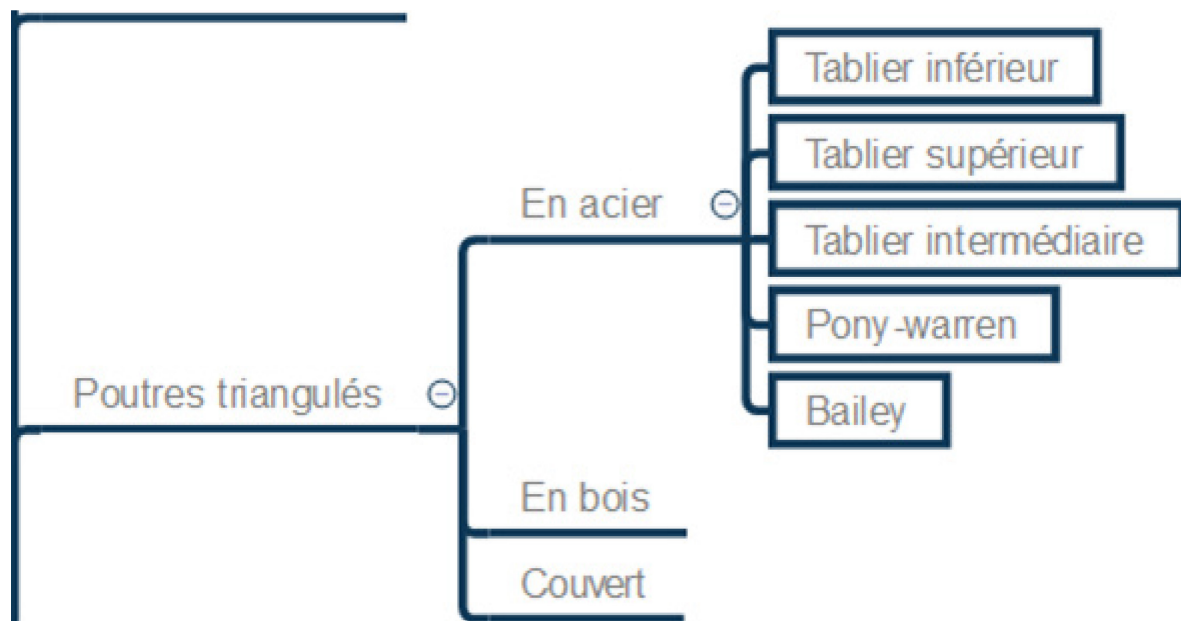


Figure 14 - Modèle conceptuel de données (suite 2)

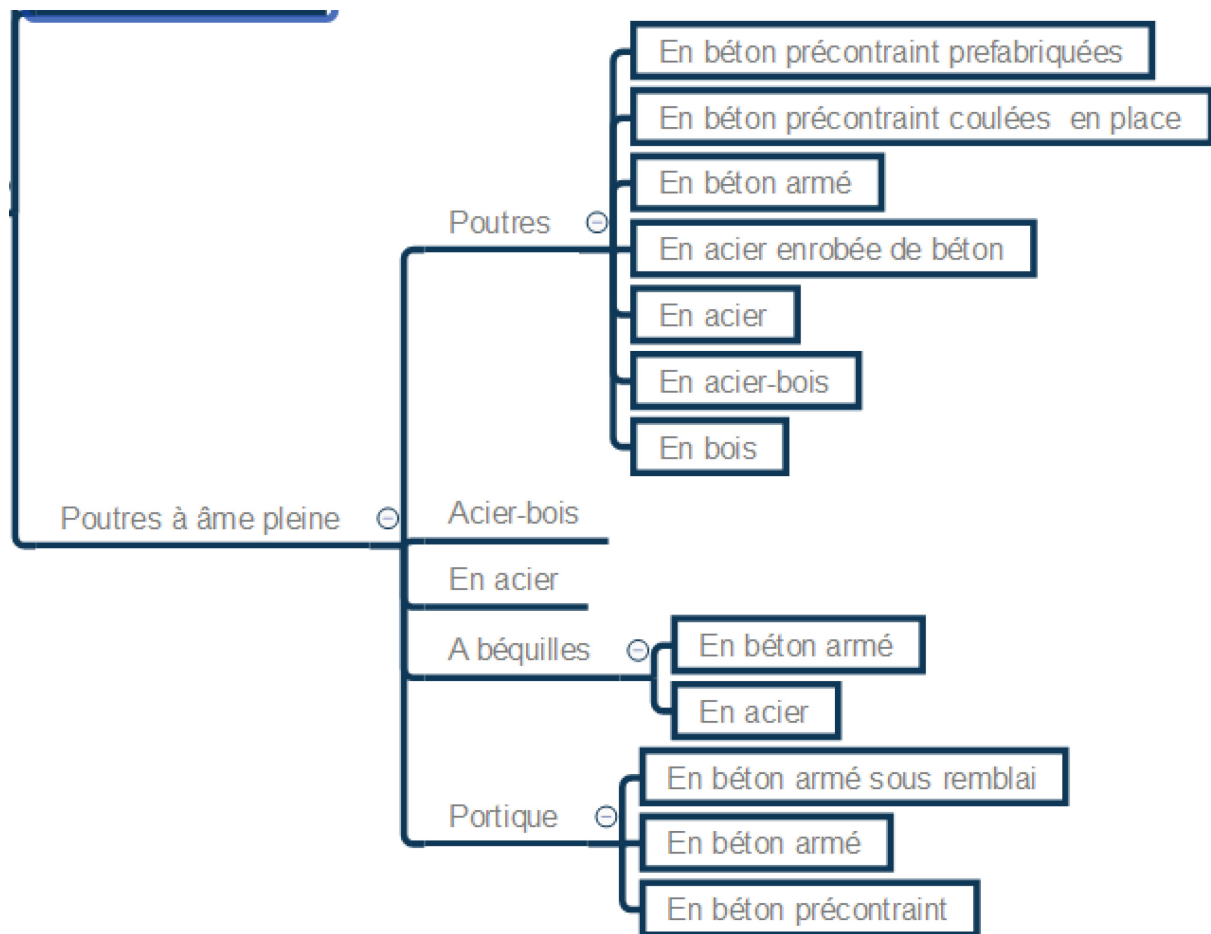


Figure 15 – Modèle conceptuel de données (suite 3)

3.4.2 Modèle logique

À partir du modèle conceptuel de données, nous avons établi un modèle logique de données illustré à la Figure 16. Pour ce faire, nous avons recensé tous les éléments du réseau routier ainsi que leurs attributs. Ensuite, nous avons conservé seulement les éléments importants qui doivent être représentés par une classe conformément au langage de modélisation (UML). Les éléments restants sont représentés sous forme d'attributs. Par exemple, le pont est représenté par une classe et la poutre, la dalle, le tablier et la béquille sont les attributs de la classe pont. La route est représentée par une classe. Le tronçon, la section et la sous-route sont des attributs de la classe route. La classe pont possède une trentaine d'attributs dont :

- un identifiant,
- une année de construction,

- un obstacle à traverser,
- des coordonnées de localisation,
- une géométrie.

La Figure 16 présente aussi les relations existantes entre les entités (élément du modèle logique). Chaque relation peut avoir un nom et une cardinalité indiquant les nombres minimaux et maximaux d'apparitions de l'entité dans la relation. Par exemple, selon notre modèle, une section traverse un ou plusieurs territoire(s) et un pont se trouve dans une section et une section peut ne pas contenir aucun pont. Enfin nous avons créé nos tables de la base de données à partir de classes jugées suffisamment importantes, chaque classe retenue en tant que telle devient une table. Notons que chaque attribut géométrie de classes route et pont permet de créer le graphe représentant le réseau routier.

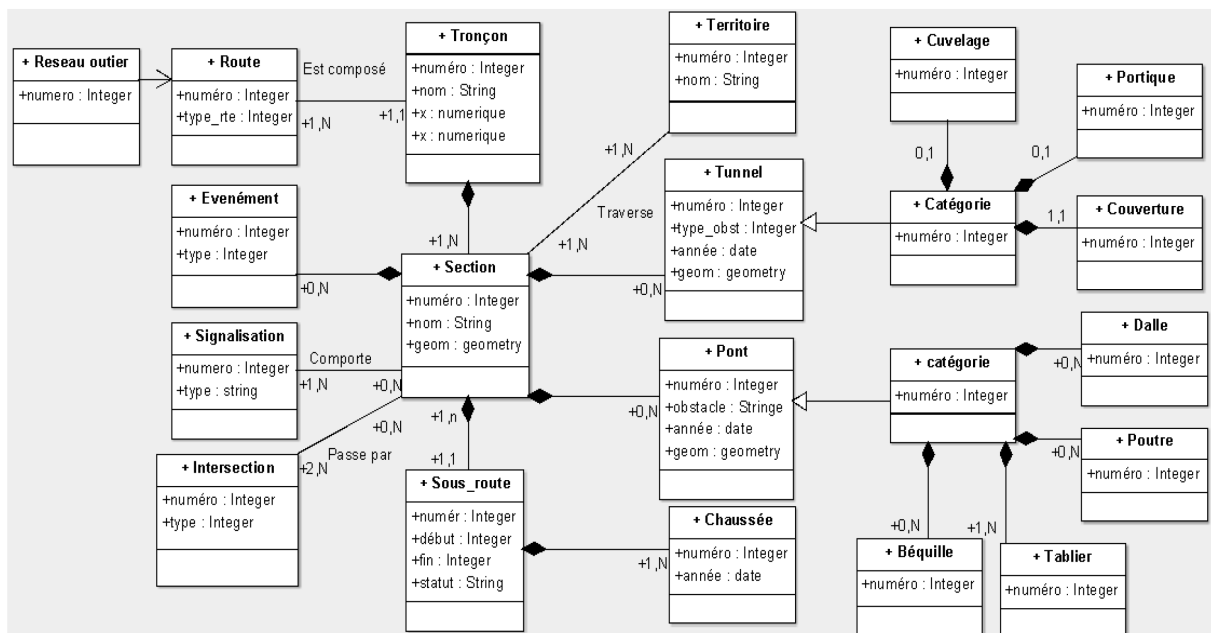


Figure 16 – Modèle logique de données

3.4.3 Intégration des données

L'intégration de données est une phase où nous explorons nos données et nous nous les approprions afin de pouvoir élaborer un plan qui nous permet d'atteindre nos objectifs. Une fois les données analysées, il faut les stocker dans une base de données. Cette dernière est la pierre angulaire de notre système. Cependant, il s'agit de modéliser les données afin qu'elles puissent être intégrées à

la base de données. Les composantes de cette base de données sont des données géographiques portant sur les routes et les ponts.

Les données utilisées dans ce projet proviennent de différentes sources. Les données sur les routes et les ponts sont obtenues à partir du site Web du MTQ par le biais de la cartothèque de l'Université de Sherbrooke qui détient une licence d'accès. Cependant il manque des informations sur les limites de vitesse et sur certains ponts. C'est ainsi que des données complémentaires portant sur les ponts sont fournies par le centre de recherche en génie parasismique et en dynamique des structures (CRGP), et celles portant sur les limites de vitesse proviennent du site OpenStreetMaps. Enfin, il faut signaler que les données sur les ponts obtenues du MTQ et du CRGP ne sont pas identiques, certains ponts ne sont pas répertoriés ainsi que certains attributs décrivant les ponts.

Le problème qui s'est posé est celui de savoir comment lier les données des ponts adéquatement à celles des routes pour faire une analyse (le calcul d'itinéraire) commune à l'ensemble des données. Comme les deux jeux de données proviennent de deux sources n'ayant pas un attribut commun permettant la liaison de ces données, il a fallu surmonter ce problème par le biais de la superposition des couches basée sur les coordonnées des entités (les routes et les ponts) qui consiste à afficher les couches des routes puis les couches des ponts au-dessus de ces dernières.

Quand nous sommes entrés en possession des données sur les ponts, nous avons constaté également un problème de différence entre les données sur les ponts fournis par le CRGP (appelons ces données DC) et les données recueillies à partir du site web du ministère (appelons ces données DM). Les DC sont incomplètes, certains ponts ne figurent pas dans celles-ci, aussi il a fallu rajouter la différence entre les DC et DM aux DC. Par contre, les DM sont aussi incomplètes au niveau d'attributs descriptifs. En effet, un nombre important d'attributs descriptifs dans leur structure (schéma) sont inexistantes. Pourtant, ces attributs sont présents dans les DC. Pour obtenir un affichage uniforme au moment de consultation des données de ponts, nous rajoutons les attributs manquants, dans le cas de DM, via le code, mais sans leur attribuer de valeur. Dans l'application, nous présentons les ponts provenant des DC et DM par deux couleurs différentes.

3.5 Implémentation

Dans notre application, une carte est formée de couches. Nous avons cinq couches de données. Ces couches concernent l'arrière-plan, deux couches pour les routes, notamment la gestion de routes et

le calcul d'itinéraire et deux autres couches pour les ponts provenant des DC et DM. Il faut préciser que certaines données de ponts sont dans le format CSV et qu'on ne peut pas les importer dans la base de données sans les convertir. Pour conserver toutes les informations, nous utilisons le SGBD PostGIS/PostgreSQL.

3.5.1 Préparation des données

Les données obtenues de différentes sources étant structurellement différentes, elles n'étaient pas prêtes à l'utilisation. Une adaptation aux besoins du projet était donc indispensable. Par exemple, nous devions :

- transposer toutes les données du système de coordonnées « *European Petroleum Survey Group* (EPSG) : 3798 » au système de coordonnées « EPSG 4326 » ; au départ les données sont dans le format du système de coordonnées NAD83 / MTQ Lambert (EPSG : 3798), mais pour visualiser ces données avec la carte de fond de OpenStreetMap que nous utilisons, elles doivent être dans le système de coordonnées EPSG : 4326 ;
- convertir les données sur les ponts obtenues du Ministère du format CSV (non exploitable par GeoServer) au format shapefile. Ce dernier format nous permet d'importer directement les données dans la base de données PostgreSQL/PostGIS ;
- découper des données via un calque pour n'extraire que les données couvrant la ville de Québec, étant donné que nous nous intéressons seulement à cette dernière ;
- extraire des données téléchargées du site Web de MTQ les ponts manquants, à partir d'un fichier shapefile dans lequel ils sont reçus, et les ajouter aux données géographiques des ponts de la ville de Québec fournies par le CRGP.

Tous ces traitements ont été réalisés dans les logiciels QGIS [73] et ArcMap [74].

3.5.2 PostGIS/PostgreSQL

Une fois les données converties au même format et adaptées à nos besoins, nous les intégrons à la base de données. PostGIS est un SGBD spatial développé par la société canadienne « Refrations Research ». C'est une extension de PostgreSQL écrite en C. Elle permet de stocker et de manipuler des objets géographiques. Cette extension ajoute à PostgreSQL des fonctionnalités lui permettant de manipuler les données géographiques. Rappelons que les données géographiques ont, en plus des attributs de base, des composantes de localisation, de géométrie et de métadonnées sur le

service proposé, par exemple les métadonnées de la couche de ponts sont les noms des attributs. Les manipulations des objets géographiques obéissent aux normes de l'*Open Geospatial Consortium (OGC)*.

La Figure 17 présente l'architecture de PostGIS. Les modules essentiels à son fonctionnement adéquat sont :

- Liblwgeom : la bibliothèque permettant de gérer les opérations qui ne sont pas directement liées à la base de données telles que la détection et correction de la corruption de mémoire, le calcul de distance entre deux points et le calcul de l'intersection entre deux géométries (point, ligne ou polygone) ;
- LibXML : la bibliothèque pour les conversions de XML à KML ou GML qui sont des formats de stockage des données géographiques ;
- Geos : le module qui s'occupe des opérations manipulant de données géographiques ;
- GDAL : la bibliothèque comportant des fonctionnalités « raster » permettant de manipuler des images ;
- Proj4 : composant pour gérer les différents systèmes de projection.

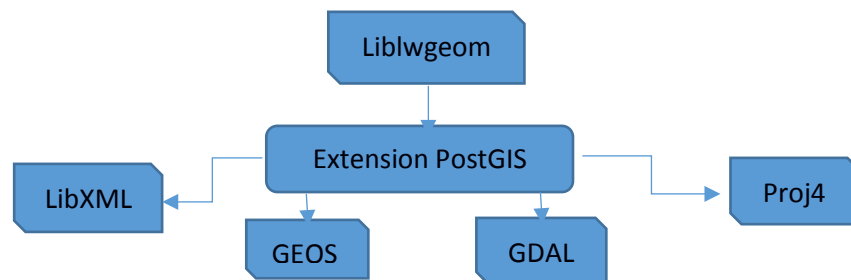


Figure 17 - Architecture de PostGIS

Pour créer notre base de données à partir de données obtenues, nous avons utilisé PgAdmin III qui est une interface graphique facilitant l'administration de la base de données PostgreSQL et PostGIS. Il s'agit d'un outil gratuit, développé en C++, assistant l'utilisateur dans l'écriture des requêtes SQL. La Figure 18 donne un aperçu de l'interface de pgAdmin.

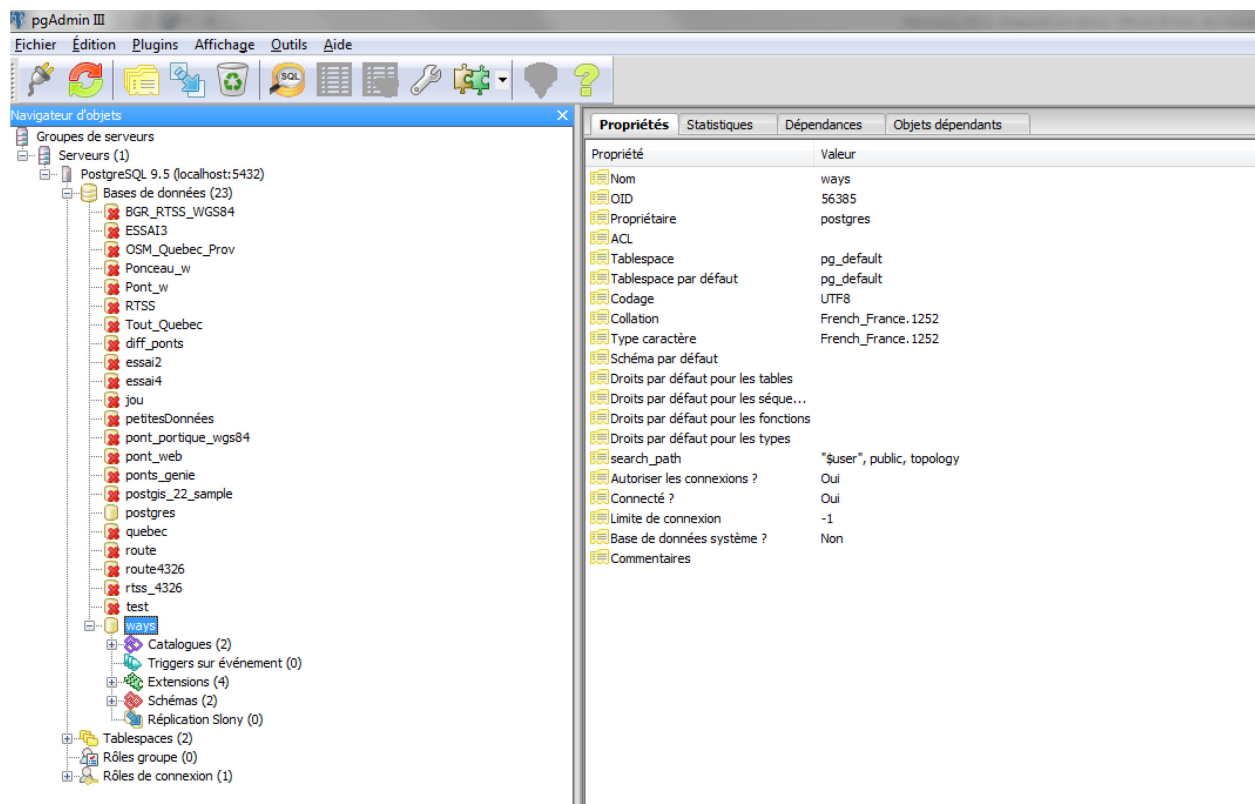


Figure 18 - Interface de pgAdmin

Les étapes dans la création commencent par la création de la base de données. Une fois celle-ci créée, les extensions nécessaires peuvent être ajoutées :

- activation de PostGIS via la commande « **CREATE EXTENSION PostGIS** » qui permet d'installer les modules Liblwgeom, LibXML, Geos, GDAL et Proj4,
- configuration du module permettant la création de la topologie qui est le graphe nécessaire à pgRouting via la commande « **CREATE EXTENSION postgis_topology** »,
- Installation de pgRouting qui est le module de calcul d'itinéraire via la commande « **CREATE EXTENSION pgrouting** ».

Une fois la base de données créée et toutes les extensions installées, il faut importer les données. Le chargement de données est effectué via l'interface de pgAdmin comme illustré à la Figure 19. Le chargement de données est effectué à partir de l'onglet « plugin » ainsi qu'en utilisant les outils « osm2pgsql » et « shp2pgsql » qui sont utilisés respectivement à convertir du format de Open

Street Map (OSM) en SQL et du format shapfile en SQL. Les données chargées dans la BD sont les routes et les ponts à partir des fichiers shapfile et OSM.

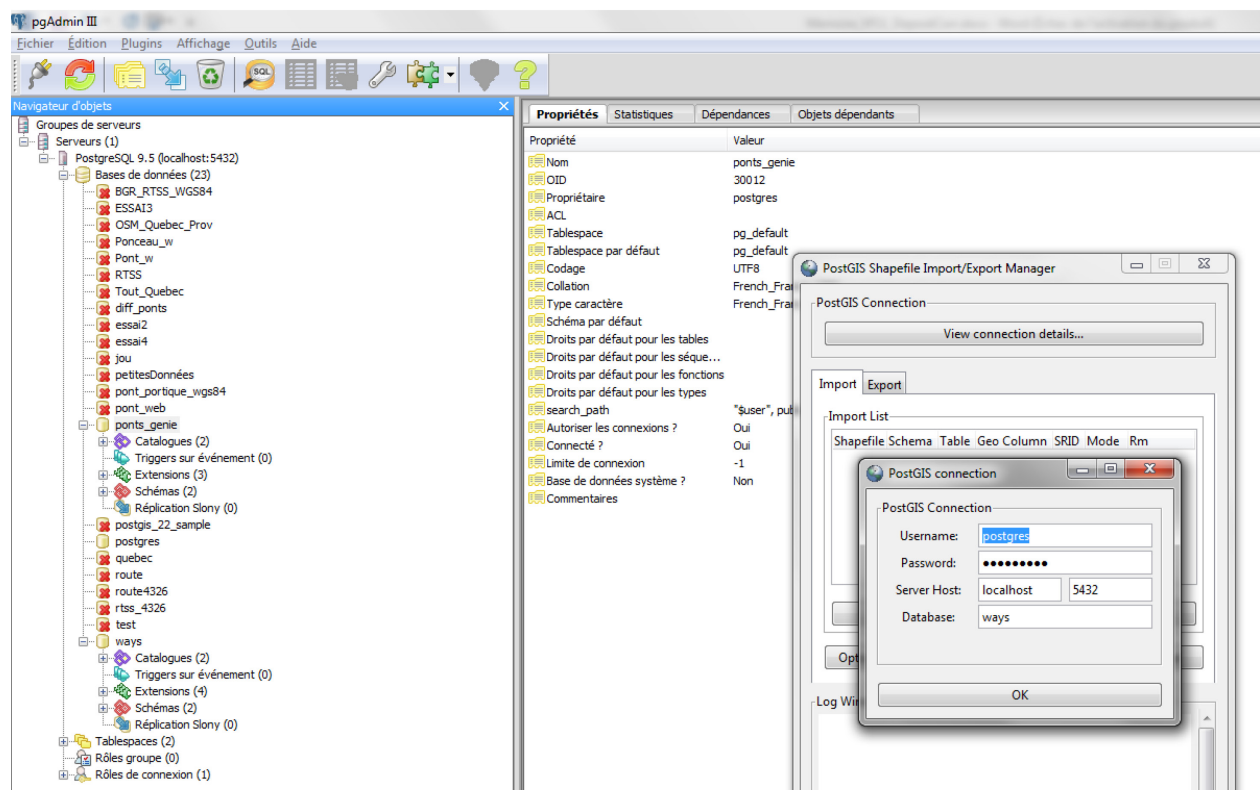


Figure 19 - Résumé des outils utilisés

Une fois les données chargées, celles-ci peuvent être consultées par l'interface pgAdmin. Les Tableaux 5, 6, 7 et 8 présentent les schémas des tables de notre base de données. Il faut noter que chaque couche correspond à une table dans la base de données. Ces schémas sont essentiellement les attributs décrivant les caractéristiques des routes et des ponts, ainsi que leurs coordonnées de localisation et leurs formes géométriques. Les Tableaux 5 et 6 représentent respectivement les données de routes issues de MTQ et d'Open Street Map (OSM). Puis les Tableaux 7 et 8 représentent les données DC et DM décrites dans la section « Analyse de données ». Dans les tables, le champ « Objet » désigne l'objet représenté par la table, le champ « projection » décrit comment les données géographiques sont représentées par rapport à leur emplacement sur la surface terrestre et enfin le champ « attributs » décrit les attributs de l'objet de la table. Le champ description n'existe pas dans la table réelle de la base de données, Il est présenté uniquement pour des fins de documentation. Ces tableaux présentent un aperçu des schémas des tables de la base de données. Les schémas complets sont fournis en annexe A.

Tableau 5 - Réseau routier (Source de données, Cartothèque de l'UDES et ministère des Transports, Division de la géomatique, 2015)

Objet	Projection	Attributs	Description
Route1 (560 enregistrements)	WGS 84 (EPSG:4326)	Route	Numéro de la route
		Tronçon	Numéro du tronçon
		Section	Numéro de la section
		Longueur	Longueur de la section de la route

Tableau 6 - Réseau routier (Source de données, Open Street Map)

Objet	projection	Attributs	Description
Route2 (123883 enregistrements)	WGS 84 (EPSG:4326)	Gid	Identifiant de l'enregistrement
		Length	Longueur de segment
		Source	Début de segment
		Target	Fin de segment
		Cost	Coût de l'allée
		The_geom	Géométrie du segment

Tableau 7 - Ponts (Source de données, initiateurs du projet)

Objet	projection	Attributs	Description
Pont (127 enregistrements)	WGS 84 (EPSG:4326)	NO STRUCTURE	Identifiant de l'enregistrement
		CLASSE DE ROUTE	Classe De Route
		TYPE STRUCTURE	Type Structure
		Nom	Nom du pont
		OBSTACLE	L'obstacle traversé

Tableau 8 - Ponts (Source de données, Site web du Ministère des Transports, Division de la géomatique, 2015)

Objet	projection	Attributs	Description
Pont2 (55 enregistrements)	WGS 84 (EPSG:4326)	Gid	Identifiant de l'enregistrement
		Longitude	Coordonnées de localisation du pont
		Latitude	
		Name	Nom du pont
		Geom	Géométrie du pont

3.6 Serveur cartographique

Le rôle d'un serveur cartographique est de créer des cartes à partir des données géographiques stockées dans la base de données. Il existe plusieurs serveurs cartographiques dont les plus populaires sont, entre autres, GeoServer, Mapserver et CARTOWEB3. Pour ce projet, nous avons choisi GeoServer pour sa maturité et sa popularité. GeoServer comprend aussi un serveur Web qui assure la communication avec le client.

Le rôle du serveur est de recevoir et traiter les requêtes des clients. Pour traiter celles-ci, il cherche dans ses ressources (base de données) les informations nécessaires à la construction de la carte, et selon la requête (WMS ou WFS), il envoie en réponse au client le résultat soit sous forme de carte (image), dans le cas d'une requête de type WMS, soit sous forme de données brutes, quand il s'agit d'une requête de type WFS. Dans le cas d'une requête de type WMS, le rôle du client consiste à afficher la carte, alors que dans le cas d'une requête de type WFS, le client effectue d'abord le traitement nécessaire à la construction de la carte avant de l'afficher.

3.6.1 Concept de GeoServer

GeoServer est un serveur cartographique écrit en Java qui nécessite l'installation d'un JDK 1.4 ou supérieur. Il est basé sur GeoTools, qui est une bibliothèque qui fournit des fonctions permettant de se connecter à la base de données, de manipuler la géométrie et de faciliter l'interopérabilité des applications de type SIGWeb.

GeoServer reçoit des requêtes formulées en JavaScript, les interprète et renvoie des résultats sous forme de données vectorielles (WFS) ou matricielles (WMS). Il est compatible avec des nombreuses bases de données et supporte plusieurs formats de données cartographiques.

GeoServer possède une arborescence de fichiers simple et bien structurée. Pour représenter l'information géographique.

Il y a d'abord les fichiers « **couches** » qui permettent de représenter un ensemble d'objets du monde réel sur la carte. Les couches sont des données vectorielles ou matricielles contenant les objets représentés dans la carte tels que les ponts [68]. Il est possible de superposer plusieurs couches dans une même interface ;

Il y a ensuite le dossier « **entrepôt** » qui permet de connecter à GeoServer une source de données (un fichier ou une table d'une base de données) contenant des ressources. Ces ressources peuvent être un fichier, un répertoire ou une table d'une base de données. Sa fonction est de définir les paramètres de connexion une seule fois pour toutes les données plutôt que de le faire pour chaque donnée d'une source différente ;

Enfin il y a un dossier appelé « **espace de travail** » qui est un conteneur d'objets spatiaux où les couches existantes sont sauvegardées. Dans le script on se réfère toujours à l'espace de travail pour faire appel à une couche. Il permet en même temps d'éliminer les ambiguïtés qui peuvent exister entre les couches. Par exemple deux couches ayant le même nom sont différenciées par le nom de l'espace de travail de chacune.

Rappelons que les couches sont contenues dans l'entrepôt et ce dernier est dans l'espace de travail.

3.6.2 Implémentation

Le travail que nous avons fait à ce niveau concerne principalement l'implémentation de programmes (scripts) pour l'importation des données. Rappelons que le but est de créer une carte globale ayant des couches de ponts et de routes et permettant de calculer un itinéraire et d'afficher cet itinéraire sous forme d'une nouvelle couche. Pour cela nous devons créer des couches les ponts, les routes et l'itinéraire calculé dans GeoServer afin d'encapsuler les données, et de créer également des couches pour les routes, les ponts et l'itinéraire au niveau du client pour représenter les résultats reçus du serveur afin de les afficher. Ainsi ces programmes du côté du serveur, que nous présentons, concernent la création des couches, l'appel des algorithmes existants dans pgRouting et la liaison

des coordonnées de points sélectionnés par l'utilisateur aux coordonnées stockées dans la base de données, car les points sélectionnés par l'utilisateur ne sont pas précis. Mais avant tout nous présentons le fonctionnement de GeoServer. En particulier, nous montrons comment créer et visualiser une couche dans Geoserver. Ainsi, une fois GeoServer installé, pour qu'il soit prêt à l'utilisation, le serveur doit être lancé via le fichier startup.bat.

Une fois démarré, on accède GeoServer via l'interface Web d'administration à l'adresse : *http://<host>:<port>/geoserver* et ainsi effectuer des manipulations comme l'ajout de données, le paramétrage, etc. La figure 20 présente l'interface de GeoServer une fois que la connexion est établie.

Il est alors possible de créer des espaces de travail, des entrepôts et des couches. Il est également possible de prévisualiser les couches. Il faut préciser que la création de couches est faite à l'aide de pointeur vers la base de données.

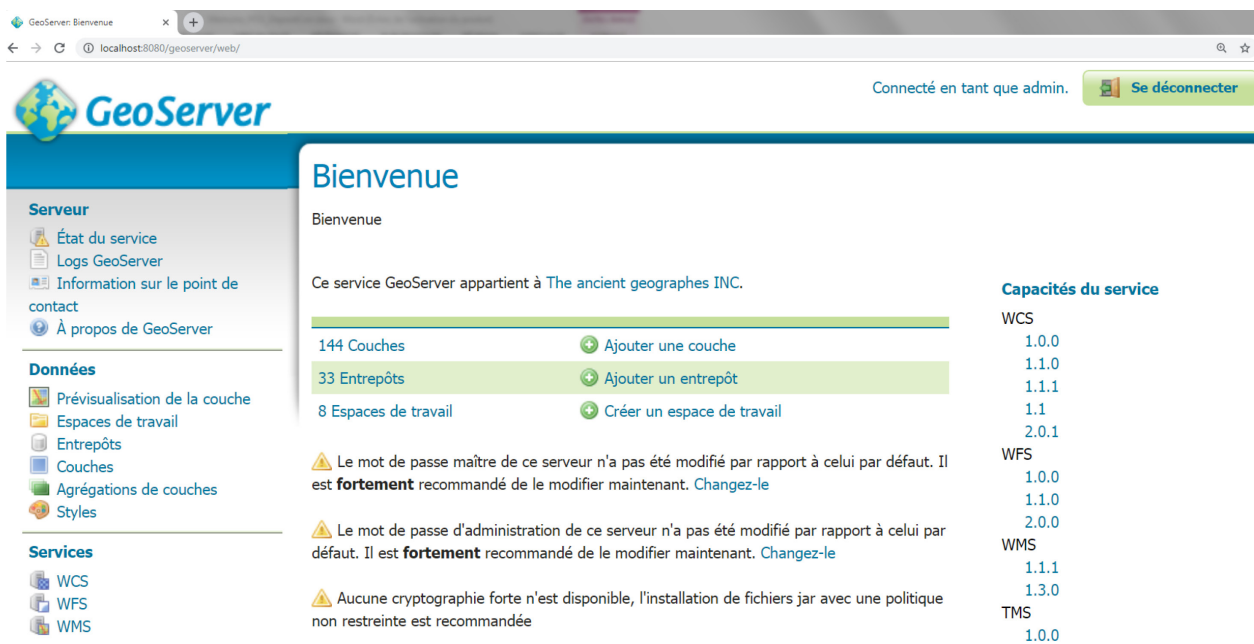


Figure 20 - Interface de GeoServer

Pour la **création de couches dans GeoServer**, une fois GeoServer bien configuré, il devient possible d'afficher les cartes. Pour y parvenir, il faut créer les couches. La création d'une couche implique plusieurs étapes. Tout d'abord, créer un espace de travail et un entrepôt à partir du module

de données de GeoServer ; la source de données est une table de la base de données PostGIS. Par la suite, créer une couche en sélectionnant dans le menu de gauche la commande « couche » et choisir une ressource (à partir du menu déroulant) qui n'est autre qu'un des entrepôts créés précédemment. Enfin, vient la configuration des paramètres nécessaires à l'affichage de la couche (par exemple, définir le système de coordonnées de référence de la couche). Ces étapes sont illustrées par la Figure 21.



Figure 21 - Étapes de création d'une couche

Une fois que les données sont chargées dans le serveur et que les couches sont créées, on peut visualiser les couches à partir de l'interface pgAdmin tel qu'on peut le voir à la Figure 22.

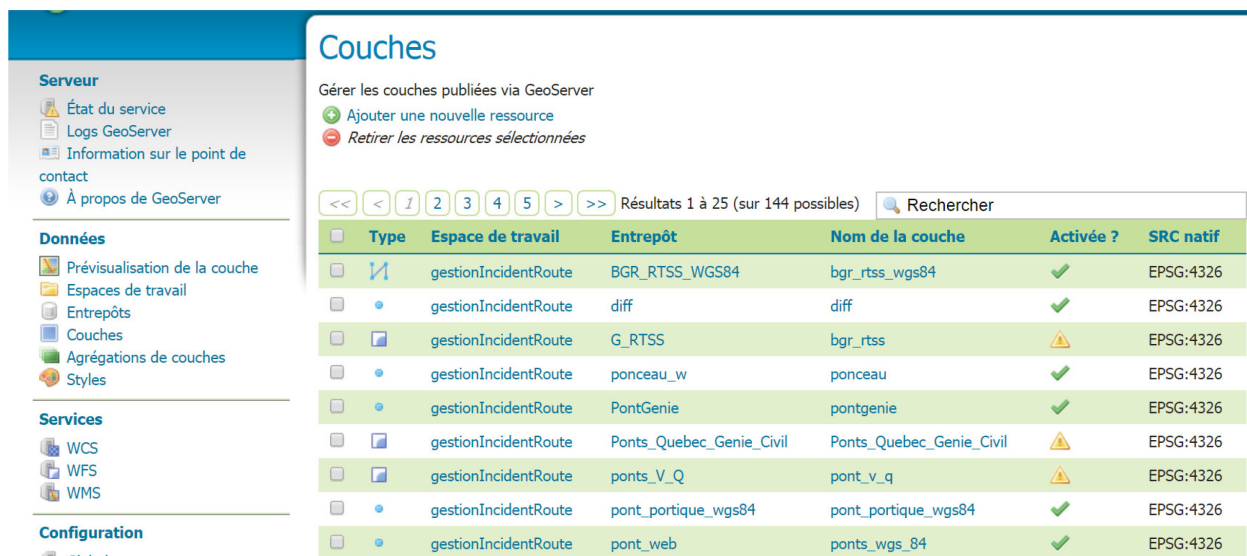


Figure 22 – Vue de nos ressources dans GeoServer

3.6.3 Code Serveur

Afin de simplifier toutes les étapes que nous venons d'aborder, nous avons développé une fonction PostgreSQL adaptée à nos structures de données qui permet d'éviter les requêtes SQL complexes et répétitives. Elle permet aussi d'éviter les attaques par injection SQL (insertion de script compromettant dans la requête SQL). Cette fonction est écrite en PL/pgSQL, et prend également en paramètres les coordonnées des points d'origine et destination (issues de la sélection de l'utilisateur sur la carte) et cherche les points correspondants dans la table de routage. Elle fait aussi appel à l'algorithme de routage A* calculant l'itinéraire. Cette partie sera abordée en détail dans le chapitre suivant.

Le script de la Figure 23 présente une partie de la fonction effectuant la recherche d'identifiants des deux points dans la table représentant les routes du réseau routier qui sont les plus proches des coordonnées des points (x1, y1 et x2, y2) reçus du client.

```
Partie 1: EXECUTE 'SELECT gid::integer as pd FROM ' || tbl || ' ORDER BY
the_geom <-> ST_GeometryFromText("POINT(" || x1 || ' ' || y1 || ')",4326) LIMIT 1'
INTO rec; troncid:=rec.pd;

Partie 2: EXECUTE 'SELECT gid::integer as pd2 FROM ' || tbl || ' ORDER BY
the_geom <-> ST_GeometryFromText("POINT(" || x2 || ' ' || y2 || ')",4326) LIMIT 1'
INTO rec; troncid2:=rec.pd2;
```


3.7 Client

Le client est toujours l'initiateur de l'échange avec le serveur. Il envoie une requête au serveur, qui la traite et renvoie la réponse. Le client à son tour reçoit la réponse et l'affiche. Pour ce projet, le type de client choisi est un client lourd. Cela signifie qu'un navigateur Web et des extensions sont nécessaires sur l'ordinateur client. Il a donc fallu intégrer la bibliothèque openlayers3.

Le client reçoit les données à l'état brut, il doit exécuter des scripts pour créer les couches nécessaires avec ces données et les afficher sur la carte.

3.7.1 Implémentation

Dans cette section nous présentons l'implémentation du programme client. D'abord nous montrons la création de couches du côté client destinées à recevoir les résultats produits par le serveur et l'affichage de ces couches. Ensuite nous présentons les opérations nécessaires pour former la carte globale à partir des couches créées et pour afficher cette carte. Nous présentons également comment le client envoie des requêtes demandant le calcul d'itinéraire et comment il récupère et affiche les résultats. Enfin, nous discutons des problèmes rencontrés lors de l'implémentation de cette phase.

3.7.1.1 Bibliothèque Openlayers

À présent le serveur est bien configuré et prêt à l'utilisation. Les données sont dans PostgreSQL/PostGIS. GeoServer et pgRouting sont configurés, les fonctions nécessaires et les couches sont créées. Bref, tous les services sont disponibles. Le serveur n'attend que des requêtes de l'application cliente développée à l'aide de la bibliothèque Openlayers3.

a) Création d'une couche (client)

Dans le cas d'une couche vecteur, les données sont envoyées au client à l'état brut. Dans ce cas, le client doit créer une couche pour les afficher, car tous les objets affichés sur une carte doivent être dans une couche. La création d'une couche de données vecteur dans laquelle le résultat sera récupéré et affiché est réalisée comme montré à la Figure 24, dans laquelle le message à envoyer à GeoServer est placé dans une variable nommée *vectorS*. Le contenu est essentiellement formé de la requête et du format de données en réponse.

```

var vectorS = new ol.source.Vector({
  format: new ol.format.GeoJSON(),
  url: function(extent) {
    return 'http://localhost:8080/geoserver/gestionIncidentRoute/ows?service=WFS&' +
      'version=1.0.0&request=GetFeature&typename=gestionIncidentRoute:rtss_rvq&' +
      'outputFormat=application/json&srsname=EPSG:4326&' +
      'bbox=' + extent.join(',') + ',EPSG:3857';
  },
  strategy: ol.loadingstrategy.bbox
});/

```

Figure 24 – Création d’une couche

Un exemple d’une requête client OpenLayers est présenté à la Figure 25. On peut voir dans la requête que le protocole HTTP est utilisé pour assurer la communication entre le client et le serveur. La requête contient dans l’ordre :

- l’URL identifiant le serveur et la ressource : ***http://localhost:8080/geoserver/gestionIncidentRoute/ows?service=WFS&version=1.0.0,***
- le type de la requête : ***request= GetFeature,***
- le nom de la ressource (la couche ici): ***typename=gestionIncidentRoute:rtss_rvq,***
- le nom du système de projection : ***srsname=EPSG:4326,***
- la zone de la carte concernée par la requête : ***bbox=extent.join(',') ,***
- le format des données en réponse : ***outputFormat=application/json.***

```

'http://localhost:8080/geoserver/gestionIncidentRoute/ows?service=WFS&version=1.0.0&request=
GetFeature&typename=gestionIncidentRoute:rtss_rvq&outputFormat=application/json&srsname=
EPSG:4326&bbox=extent.join(',')';

```

Figure 25 – Requête via le protocole HTTP

Une fois la couche créée, elle doit être mise en forme pour l’affichage. Par exemple, si le style de la Figure 26 est appliqué à une route, les lignes représentant les routes sont affichées en jaune et la largeur de la ligne est de 6 pixels.

```

var ligne = new ol.style.Style({
    stroke: new ol.style.Stroke({
        color: 'yellow',
        width: 6
    })
});

```

Figure 26 – Création d’une couche

b) Création de la carte globale

Une fois les couches créées, il faut les regrouper afin de former une carte qui pourra être affichée. Le champ ‘**div**’ nommé ‘**map**’ de la figure 27, créé dans une portion de code html de notre application, est l’identifiant de la partie de la page Web dans lequel la carte globale sera affichée. L’organisation des couches est faite de manière à ce que la carte de base soit placée en arrière-plan, et que le reste de couches soient ajoutées par-dessus. La partie de la ville de Québec que nous souhaitons afficher au centre ‘**Center**’ de la page Web est définie par les coordonnées -71.219, 46.814. Le script ‘**zoom: 14**’ précise la taille de l’affichage. La Figure 29 illustre que la réalisation de ces tâches est établie essentiellement par trois éléments, le ‘**target**’ pour indiquer l’endroit où la carte sera affichée, le ‘**layer**’ pour spécifier toutes les couches à afficher sur la carte et le ‘**view**’ qui définit la zone de la carte qui devrait être le centre de l’affichage.

```

target: 'map',
layers: [
    new ol.layer.Group({
        'title': 'Base maps',
        layers: [
            ponts,
            vectorRTSS,
            Pont_diff ] }) ],
view: new ol.View({
    center: ol.proj.fromLonLat([-71.219, 46.814]),
    zoom: 14
}), });

```

Figure 27 – Création d’une carte

c) Affichage des informations des objets de couches

Une fois la carte globale de l'application affichée, nous passons à l'interaction des utilisateurs avec la carte présentant des données sous forme de couches. Lorsqu'un utilisateur interagit avec l'interface de l'application, il est nécessaire d'identifier la couche concernée par l'interaction. Nous appelons cette couche la couche active. La Figure 28 montre la partie du code qui permet de sélectionner la couche active. Nous identifions cette couche de données en se basant sur l'identifiant qu'on lui a attribué lors de sa création et sur le capteur d'évènements (sélection de l'utilisateur) et l'ordre des couches. Par exemple, quand un utilisateur sélectionne un pont pour consulter ses attributs, on doit pouvoir sélectionner la couche de ponts pour éviter d'afficher les attributs de la route qui porte le pont.

```
var selectInteraction = new ol.Interaction.Select({  
  layers: function (layer) {  
    if(layer.get('id') == 'pontD'){  
      return Pont_diff;}  
    else if(layer.get('id') == 'pontG') {  
      return vector; }  
    else if(layer.get('id') == 'waysl'){  
      return losm;}  
    else return null; }  
});  
map.getInteractions().extend([selectInteraction]);
```

Figure 28 – Affichage des données en couches

La figure 29 montre comment afficher les informations relatives à une route ou un pont quand un utilisateur fait une sélection. Les attributs d'objets (d'une couche) sélectionnés par l'interaction utilisateur sont d'abord récupérés dans des variables. On crée ensuite des tableaux d'objets pour y stocker les objets sélectionnés avec leurs attributs (voir figure 29). Par exemple, quand un utilisateur sélectionne un pont, je récupère toutes les informations (nom d'attributs, valeurs des attributs, etc.) liées à ce pont dans un tableau à partir de la couche de ponts.

```

var displayFeatureInfo = function(pixel, c) {
    var features = [];
    var featuresG = [];
    var featuresD = [];
    map.forEachFeatureAtPixel(pixel, function(feature, layer) {
        if(layer.get('id')=='pontG'){
            featuresG.push(feature);
        }else if(layer.get('id')=='pontD'){
            featuresD.push(feature);
        }else if(layer.get('id')=='routeR') {features.push(feature);}
    });
};

```

Figure 29 – Contrôle de l'interaction utilisateur

Une fois que la couche subissant l'action de l'utilisateur est identifiée, les informations contenues dans le tableau (décrit à la figure 29) sont récupérées et affichées, et ceci, concerne les attributs et leurs valeurs. La figure 30 montre un exemple d'affichage de propriétés d'un pont.

```

if(featuresG.length > 0){
    var message2 = '<div><h2> Pont </h2><br>'+
        '<h5>Numero:</h5>'+featuresG[0].getProperties().NO_STRUCTU+
        '<h5>Type_Structure:</h5>'+featuresG[0].getProperties().TYPE_STRUC+
        '<h5>Nom:</h5>'+featuresG[0].getProperties().NOM+
        '<h5>Obstacle:</h5>'+featuresG[0].getProperties().OBSTACLE+
        featuresG=[];
    popup.show(c, message2);
};

```

Figure 30 – Affichage de résultat à l'utilisateur

d) Récupération et affichage de l'itinéraire calculé

Quand un utilisateur veut calculer un itinéraire, il doit d'abord choisir son point de départ et son point de destination. Cette information est ensuite envoyée au serveur, et ce dernier lui retourne le résultat sous forme de données brutes. Le client doit créer les couches pour afficher le point de départ, le point de destination et l'itinéraire calculé.

Le code présenté à la Figure 31 montre comment créer les deux symboles graphiques (vecteurs d'objets) représentant les deux points d'origine et de destination sur la carte. Et comment créer une couche de type vecteur qui doit contenir le résultat de l'itinéraire calculé entre les deux points à afficher sur la carte. Il faut définir des paramètres à travers lesquels les coordonnées des deux points sont envoyées au serveur et enfin, il faut traduire les coordonnées du format de données de la page Web au format compatible avec les données stockées au serveur.

```
var startPoint = new ol.Feature();  
  
var destPoint = new ol.Feature();  
  
var lespoints = new ol.layer.Vector({  
    source: new ol.source.Vector({  
        features: [startPoint, destPoint]}  
    });  
  
    map.addLayer(lespoints);  
  
var params = {  
    LAYERS: 'gestionIncidentRoute:vue_rtss_4326',  
    FORMAT: 'image/png'  
    };  
  
var transform = ol.proj.getTransform('EPSG:3857', 'EPSG:4326');
```

Figure 31 - Création d'une couche

La Figure 32 montre le programme permettant de récupérer les coordonnées des points lors de l'interaction avec l'utilisateur (clics). À l'étape 1, on effectue des tests pour localiser les points sélectionnés par l'utilisateur, puis on marque ces points sur l'interface graphique. À l'étape 2, on projette les coordonnées des points sélectionnés du système de coordonnées « EPSG : 3857 » de la

page Web (Open Street Map) vers le système de coordonnées « EPSG : 4326 » qui est le système de coordonnées utilisé dans notre base de données. Enfin à l'étape 3, les coordonnées des points sont encapsulées dans un tableau de paramètres qui sera transmis à GeoServer.

```
1: if (startPoint.getGeometry() == null) {  
    startPoint.setGeometry(new ol.geom.Point(evt.coordinate));  
}  
else if (deffPoint.getGeometry() == null) {  
    deffPoint.setGeometry(new ol.geom.Point(evt.coordinate));  
}  
else if (destPoint.getGeometry() == null) {  
    destPoint.setGeometry(new ol.geom.Point(evt.coordinate));  
}  
2: var startCoord = transform(startPoint.getGeometry().getCoordinates());  
    var deffCoord = transform(deffPoint.getGeometry().getCoordinates());  
    var destCoord = transform(destPoint.getGeometry().getCoordinates());  
3: var viewparams = [  
    'x1:'+startCoord[0], 'y1:'+startCoord[1],  
    'x2:'+destCoord[0], 'y2:'+destCoord[1],  
    'x3:'+deffCoord[0], 'y3:'+deffCoord[1]  
    ];
```

Figure 32 – Création d'une couche (suite)

Le programme de la Figure 33 montre comment la requête, demandant l'itinéraire destinée à GeoServer, est construite et comment le résultat est récupéré et affiché. L'étape 1 envoie une requête de type WFS à GeoServer demandant l'itinéraire entre les deux points mis en paramètre dans la variable « viewparams ». Cette requête est la même du point de vue technique, que la requête de type WFS déjà expliquée dans la section « Création de couches ». Celle-ci se distingue de la requête expliquée dans la section « Création de couches » seulement par la présence des coordonnées de deux points (de départ et de destination). L'étape 2 concerne le style de la couche.

À l'étape 3, on construit la couche de données du résultat et on l'emmagasine dans la variable « result ». Le résultat est ensuite affiché sur la carte par la commande '*map.addLayer(result);*'.

```
1:  var itineraire_solution = new ol.source.Vector({  
    format: new ol.format.GeoJSON(),  
    url: function(extent) {  
        return 'http://localhost:8080/geoserver/gestionIncidentRoute/ows?service=WFS&' +  
            'version=1.0.0&request=GetFeature&typename=gestionIncidentRoute:ruEmp&' +  
            '&viewparams=' + viewparams.join(';')+  
            '&outputFormat=application/json&srsname=EPSG:4326&' +  
            'bbox=' + extent.join(',') + ',EPSG:3857';  
    },    strategy: ol.loadingstrategy.bbox    });  
2:  var styling = new ol.style.Style({  
        stroke: new ol.style.Stroke({  
            color: 'green',  
            width: 8                })  
    });  
3:  var result = new ol.layer.Vector({  
        id: 'Pg',  
        title: 'Itineraire_Pgrouting',  
        source: itineraire_solution,  
        style: styling/    });
```

Figure 33 – Création d'une couche (suite)

3.8 Difficultés rencontrées

Au cours de la réalisation de ce projet, nous avons été confrontés à des problèmes de natures diverses. Ces problèmes sont d'ordres administratifs et techniques.

D'abord nous n'avons pas trouvé d'articles traitant de sujets similaires au nôtre, c'est-à-dire des articles qui font la gestion des infrastructures routières via le WEBGIS et qui font en même temps un calcul d'itinéraires.

Il faut aussi souligner la difficulté rencontrée, pour l'obtention des données sur les infrastructures routières du Québec. Cette étape a nécessité un certain délai à cause des procédures administratives.

Un autre problème rencontré était la difficulté à publier (afficher) les données sur les routes et les ponts en utilisant le service WFS. GeoServer doit connaître le système de coordonnées de référence (CRS) des données qu'il récupère de la base de données car il est nécessaire au moment de l'affichage des données sur la carte. Or, il n'est pas demandé explicitement de renseigner le système de coordonnées de référence lors de chargement des données dans base de données. GeoServer répond donc à la requête du client par des données sans le système de coordonnées de référence, ce qui empêche tout affichage correct de la part du client. C'est ainsi que nous avons rechargé les données dans la base de données et renseigné le CRS via une option (disponible en mode invite de commande et en mode graphique pgAdmin) à rajouter lors du chargement.

Chapitre 4 Calcul d'itinéraire

Notre système, en plus de permettre la gestion des infrastructures routières, permet le calcul d'itinéraire en cas de bris ou de blocage au niveau du réseau routier. Dans ce chapitre nous abordons donc comment se fait ce calcul d'itinéraire en parlant du module utilisé, de son mode de fonctionnement et des critères utilisés pour le calcul. Nous parlons aussi de son implémentation et de son intégration dans notre système via la création de vue et l'ajout des paramètres. Enfin, nous parlons des difficultés rencontrées.

4.1 Module de calcul d'itinéraire

Pour effectuer le calcul d'itinéraire, nous utilisons un module appelé pgRouting. Pour pouvoir se servir de ce module, il est impératif que les données soient traitées pour être dans la structure requise par pgRouting. Étant donné que pgRouting ne peut pas utiliser le graphe dans son état brut, nous devons le traiter en créant ce qu'on appelle une topologie qui consiste à créer des nœuds là où les tronçons de route se croisent et en assignant un identifiant au début et à la fin de chaque tronçon. Le résultat de la création de la topologie est une table de routage et un graphe bien adapté à pgRouting. Ce dernier a été créé par Sylvain Pasche de l'entreprise *Camptocamp*. Il s'appelait à la base pgDijkstra et a été étendu par l'entreprise japonaise *Orkney* puis dénommé pgRouting. Jusqu'à ce jour, pgRouting reste un projet libre et il géré par la fondation *Open Source Geospatial* (OSGeo) dont la mission est de promouvoir le développement collaboratif libre de données et de technologies géographiques. Ce module a la capacité de manipuler et d'analyser les graphes. Les algorithmes qu'il implémente sont :

- K-Shortest Path;
- Turn Restriction Shortest Path (TRSP);
- Floyd-Warshall's Algorithm;
- Johnson's Algorithm;
- Shortest Path A*;
- Bi-directional A* Shortest Path;
- Bi-directional Dijkstra Shortest Path;
- Dijkstra's shortest path algorithm.

La figure 34 fournit les composantes principales et la structure interne de pgRouting. On remarque que pgRouting est intégrable dans la base de données PostgreSQL/PostGIS. Il comporte aussi les bibliothèques GAUL, CGAL et Boost Graph. La première, GAUL

TSM, est une bibliothèque permettant de calculer les solutions du problème du voyageur de commerce. La seconde, CGAL, est une bibliothèque permettant de créer des polygones de distances parcourues basées sur le plus court chemin calculé par Dijkstra. Finalement, Boost Graph Dijkstra/A* est une bibliothèque fournissant les algorithmes Dijkstra et A* qui est une implémentation de l'algorithme du plus court chemin de Dijkstra avec une Heuristique euclidienne. Pour ce travail, nous utilisons l'algorithme du plus court chemin A* pour le calcul d'itinéraires.

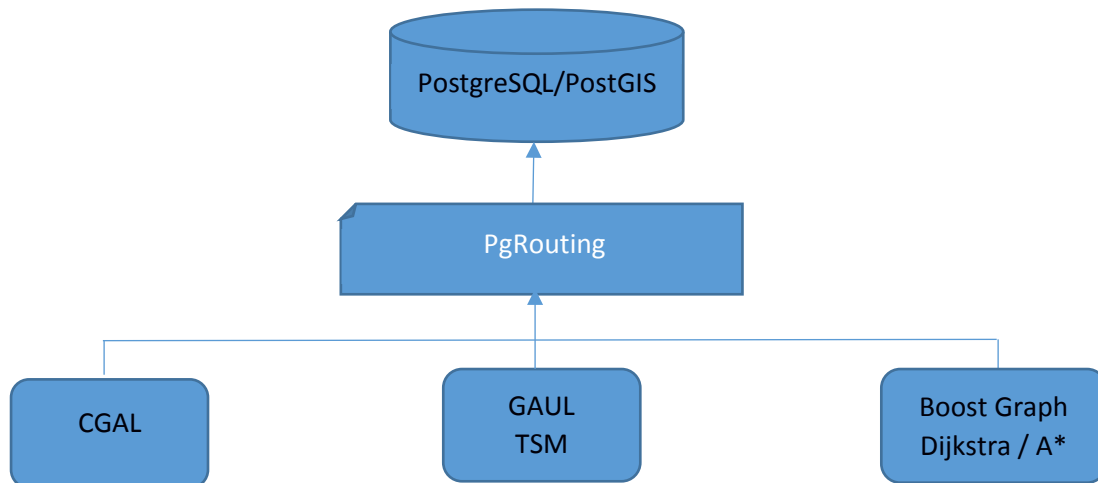


Figure 34 - Structure de pgRouting

4.2 Principe de fonctionnement

Une requête de calcul d'itinéraire envoyée à pgRouting doit spécifier les coordonnées de deux points. La première représente l'origine et la deuxième la destination. PgRouting exploite la table de données contenant les informations relatives au réseau routier (représenté par un graphe) stockées dans PostgreSQL/PostGIS pour calculer des itinéraires. D'abord, pgRouting effectue un traitement pour faire correspondre les coordonnées des deux points reçus à deux points de la table de routage (données de pgRouting). Ensuite, il adapte le graphe à ses besoins (étant donné que le graphe existant n'est pas utilisable par pgRouting d'où la création de la topologie expliquée au début de ce chapitre) et le charge en mémoire pour calculer le meilleur itinéraire entre les deux points en utilisant A* (pour notre cas). Enfin, le résultat est retourné au client. Il s'agit de l'ensemble des tronçons constituant le trajet calculé de la source à la destination.

La Figure 35 montre les données de la base de données adaptées et exploitées par pgRouting. Par exemple les attributs « sources » et « cibles » représentent les débuts et les fins de tous les

tronçons constituant le réseau routier et « cost_s » et « reverse_cost_s » représentent les coûts des tronçons dans les deux sens.

	gid [PK] bigint	class_id integer	length double precision	length_m double precision	name text	source bigint	target bigint	x1 double	y1 double	x2 double	y2 double	cost_s double precision
1	9	110	0.00023162469644033	24.9407119533494	'	66893	66894	-71.09	45.25	-71.09	45.25	1.918516304103
2	10	110	0.00159931235222965	175.247783103746	Repsher Road	66895	66896	-71.48	45.06	-71.48	45.06	13.48059870028
3	23	110	0.00112123134544204	86.7410828383279	Rue Robitaille	66897	66898	-71.30	46.81	-71.30	46.81	10.84263535479
4	30	102	0.000806176258641215	62.0747301573924	'	45277	66899	-71.29	46.79	-71.29	46.79	4.774979242876
5	31	108	0.00167523717723048	149.879889178405	P13760	66900	66901	-71.01	46.81	-71.01	46.81	11.52922224449
6	33	110	0.00156315067731358	129.249961657669	Rue Cardinal	66902	66903	-71.26	46.81	-71.26	46.81	9.942304742897
7	37	110	0.000825597971172274	69.319387214964	Rue de l'Anse	25461	66904	-71.27	46.75	-71.27	46.75	5.332260554997
8	44	108	0.000723195160382671	80.3850808868564	Côte Joyeuse	66905	66906	-71.83	46.88	-71.83	46.88	3.215403235474
9	46	110	0.000426488323399416	38.7879596112951	Rue Wolfe	66907	66908	-71.17	46.81	-71.17	46.81	2.983689200868
10	58	112	0.000582684974918404	50.7527138214489	'	61371	66909	-72.75	46.52	-72.75	46.52	3.904054909342
11	59	102	0.000552152334057043	52.1418174646357	'	66910	66911	-71.21	46.85	-71.21	46.85	4.010909035741
12	74	112	0.000374692527289242	29.2345956118548	'	66912	66913	-71.19	46.85	-71.19	46.85	2.248815047065
13	76	107	0.000274779275046843	20.3418850789773	'	66914	6059	-71.07	48.43	-71.07	48.43	1.564760390690
14	80	119	0.000348586990574262	31.9855619162128	'	45251	66915	-71.27	46.78	-71.27	46.78	2.460427839708
15	1249	303	0.00820444054460664	914.077941377876	Depot Road	66916	22547	-69.75	46.94	-69.74	46.93	70.31368779829
16	1250	303	0.00268037033635214	205.918242326183	Realty Road	66917	25295	-70.01	46.59	-70.01	46.59	15.83986479432
17	2411	110	0.0015277749114403	124.399730584546	Rue Rina-Lasnier	24149	24148	-71.17	46.78	-71.17	46.78	9.569210044965
18	2412	110	0.00072643617750016	64.0239936817033	Rue Rina-Lasnier	24148	66918	-71.17	46.78	-71.17	46.77	4.924922590900
19	2413	202	0.000784320578586411	81.4459110585023	'	66919	66920	-71.22	46.75	-71.22	46.75	6.265070081423
20	2615	106	0.00137262558623944	122.526416590681	Route du Président-Kennedy	66921	66922	-71.13	46.76	-71.13	46.76	9.425108968513

Figure 35 - Données de calcul d'itinéraire

4.3 Solution de calcul d'itinéraire

Le principe du calcul est tel qu'à chaque tronçon de route, on y associe des valeurs (des paramètres donnés à pgRouting) que sont la distance, la vitesse et la priorité. Plus la valeur de la priorité d'un tronçon est petite, plus le coût est faible. Par conséquent l'algorithme de calcul d'itinéraire choisit le tronçon en question dans son calcul. Les priorités sont affectées aux tronçons de la manière suivante :

- Autoroute = 0.3,
- Nationale/régionale = 1,
- Collectrice = 4,
- Locale = 5.

La valeur 0,3 est attribuée à l'autoroute pour augmenter fortement sa priorité, sachant qu'elle ne comporte pas d'arrêts. L'attribution de la valeur 1 à la route nationale est due à la présence d'arrêts. Enfin les valeurs 4 et 5, affectées à la route collectrice et à la route locale, ont pour but de les éviter le plus possible à cause de nombreuses restrictions (feux de circulation, arrêts, limitations des vitesses, etc.) qui leur sont associées. Dans un contexte opérationnel, ces valeurs seraient fournies par l'utilisateur.

Ces valeurs sont utilisées dans une formule pour calculer le coût du tronçon. Pour aller d'un point A d'un tronçon à un point B d'un autre tronçon, on évalue le coût de chaque tronçon menant à la destination et on choisit le tronçon ayant le plus faible coût. Le coût final de l'itinéraire est la somme des coûts des tronçons retenus qui constituent l'itinéraire du point d'origine jusqu'à la destination.

La formule pour le calcul du coût est :

$$\text{Coût} = (\text{distance} / \text{vitesse}) * \text{priorité} \quad (4-1)$$

- la **distance** est la mesure de la distance d'une portion de route parcourue,
- la **vitesse** est la vitesse permise dans un tronçon de route,
- la **priorité** est une valeur attribuée selon le type de la route (Autoroute, nationale, collectrice ou locale). Cette valeur constitue un critère de choix dans le processus de calcul d'itinéraire. C'est-à-dire que pendant la sélection des routes, elle attribue à l'autoroute, à la route nationale, à la route collectrice et à la route locale, une préférence dans un ordre décroissant.

4.4 Implémentation

Nous devons déployer pgRouting pour calculer d'itinéraires. Pour cela, nous créons d'abord une fonction qui fait appel à l'algorithme A* de pgRouting pour calculer l'itinéraire, puis nous créons une vue qui fait appel à cette fonction dans GeoServer et enfin nous paramétrons cette vue.

4.1.1 Création d'une fonction de calcul d'itinéraire

Pour calculer l'itinéraire, il est nécessaire d'avoir un script qui traduit les coordonnées de deux points (le point de départ et le point de destination) transmis par l'utilisateur (à partir d'une page Web) en coordonnées correspondants aux données de notre base données exploitées par pgRouting. Pour mieux optimiser le calcul, il faut également restreindre le champ de recherche de l'itinéraire à une zone relativement petite qui comprend le point de départ et le point de destination. Étant donné que GeoServer joue le rôle d'intermédiaire entre pgRouting / base de données et le client (page Web), chaque sélection du client correspond à des requêtes que GeoServer doit effectuer à l'endroit de pgRouting / base de données. Pour ce faire, nous créons une vue pour permettre au client de communiquer avec pgRouting / base de données par l'intermédiaire de GeoServer. Sans cette vue la communication entre le client et pgRouting / base de données ne serait pas possible.

La partie du code présentée à la Figure 36 sert à rassembler les informations nécessaires au calcul d'itinéraire et à faire appel à pgRouting pour l'exécution de l'algorithme A* qui effectue le calcul. Le *ST_Reverse(the_geom)* permet de tenir compte du cas de routes à sens unique. Par souci d'optimisation, on utilise la fonction *ST_buffer(the_geom,'|| buff ||')* pour limiter la recherche sur une zone bien définie dans la variable 'buff' qui est calculée dans la fonction qu'on a écrite. Par exemple, à la ville de Sherbrooke, si on veut calculer le trajet optimal pour aller du parc Jacques Cartier à l'Université de Sherbrooke, il va de soi qu'on exclut la zone de Fleurimont de la zone de recherche.

```
sql := 'SELECT gid, the_geom, pgr_astar.cost, source, target, ST_Reverse(the_geom) AS
flip_geom
FROM'
'pgr_astar("SELECT gid as id, source::int, target::int, '
|| tbl || '.cost_s' || tbl || '.maxspeed_forward *' || tbl || '.priorite as cost, x1, y1, x2, y2,
reverse_cost_s' || tbl || '.maxspeed_forward *' || tbl || '.priorite as reverse_cost FROM '
|| quote_ident(tbl) || ' WHERE the_geom @ (SELECT ST_buffer(the_geom' || buff || ') FROM '
|| quote_ident(tbl) || ' WHERE gid=' || tgid || ' AND gid!=' || troncid || ' AND gid!=' || troncid2 || ')",'
|| source || ', ' || target
|| ' true, true), '
|| quote_ident(tbl) || ' WHERE id2 = gid ORDER BY seq';
```

Figure 36 – Code d'exécution de A*

4.1.2 Création d'une vue SQL

La vue SQL est le moyen par lequel on peut faire appel à pgRouting à partir de GeoServer. La figure 36 présente une requête SQL faisant appel à pgRouting. Cette requête se trouve dans une fonction et la fonction est intégrée dans PostGIS/PostgreSQL. La fonction est appelée par la vue comme présenté à la figure 37. L'objectif principal de la vue ici est de fournir au client un accès au service proposé sous forme d'une couche de type Web Map Service (WMS) ou de type Web Feature Service (WFS). La vue (voir Figure 37) permet également de transmettre les paramètres, reçus du client par GeoServer, à pgRouting pour calculer l'itinéraire.

La Figure 37 montre l'exécution de notre fonction à travers la « vue SQL ». Cette fonction fait appel à pgRouting pour calculer d'itinéraire. Elle prend en argument la table de routage de la base de données et les paramètres transmis par le client. Le résultat est un ensemble de tronçons qui sera transformé en une entité géométrique unique. Cette entité peut être intégrée facilement dans une couche et affichée sur la carte.

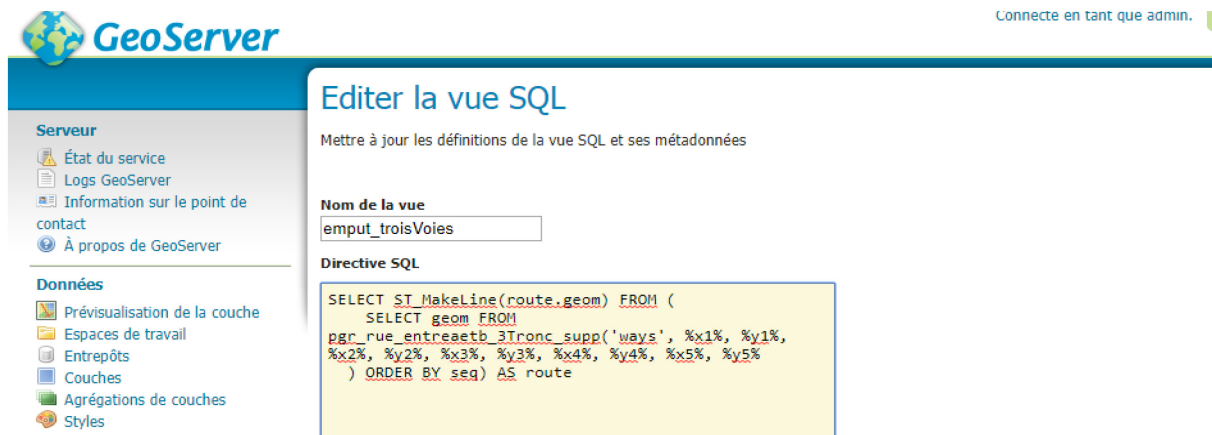


Figure 37 - Création d'une vue SQL

4.1.3 Paramétrage de la vue

Rappelons que GeoServer est l'intermédiaire reliant le client à pgRouting à l'aide de la « vue SQL ». Ainsi nous devons intégrer dans la « vue SQL » les coordonnées des points (le point de départ, le point de destination et les points désignant les tronçons de route à bloquer) transmis par l'utilisateur au serveur et qui sont nécessaires au calcul de l'itinéraire via les paramètres de la vue.

La vue prend tous les paramètres destinés au serveur depuis le client. Par exemple, à la Figure 38, il s'agit des coordonnées du point de départ, du point destination et de points représentant les tronçons de route à exclure du processus de calcul d'itinéraire. On affecte des valeurs par défaut à toutes les coordonnées attendues, celles-ci sont utilisées par défaut lorsqu'un problème de transmission de coordonnées survient. L'expression régulière sert à contrôler les valeurs (doivent être des réelles) des variables X1, Y1, X2, Y2, ..., X5, Y5 qui sont les coordonnées du point de départ, du point de destination et de points désignant les tronçons de routes à bloquer. Il faut également spécifier le type de la géométrie (de données) et le « SRID » ou « EPSG » (permettant de projeter correctement le résultat sur la carte) qui sont nécessaires lors de l'affichage sur le site client.

[Deviner les paramètres à partir du SQL](#)
[Ajouter un nouveau paramètre](#)
[Retirer la sélection](#)

<input type="checkbox"/>	Nom	Valeur par défaut	Expression régulière de validation
<input type="checkbox"/>	y1	46.82461346692173	^[lw.-]+\$
<input type="checkbox"/>	x1	-71.29680633544922	^[lw.-]+\$
<input type="checkbox"/>	y2	46.87392257597699	^[lw.-]+\$
<input type="checkbox"/>	x2	-71.19655609130858	^[lw.-]+\$
<input type="checkbox"/>	y3	46.833774295699385	^[lw.-]+\$
<input type="checkbox"/>	x3	-71.27105712890624	^[lw.-]+\$
<input type="checkbox"/>	y4	46.82555311096053	^[lw.-]+\$
<input type="checkbox"/>	x4	-71.25320434570311	^[lw.-]+\$
<input type="checkbox"/>	y5	46.78090187978415	^[lw.-]+\$
<input type="checkbox"/>	x5	-71.337833404541	^[lw.-]+\$

☐ Échapper les caractères spéciaux SQL

Attributs

[Actualiser](#)
☐ Deviner le type de géométrie et le système de référence spatial

Nom	Type	SRID	Identifiant
st_makeline	LineString ▼	4326	<input type="checkbox"/>

Figure 38 - Paramètres de la vue

Chapitre 5 Résultats

Rappelons que notre système a deux objectifs soit d'aider à la gestion des infrastructures routières et à trouver des routes alternatives en cas de bris. Nos tests reflètent ses deux objectifs. Dans ce chapitre nous présentons les tests de validation du système. Notre première série de tests concerne la gestion des infrastructures où une consultation de toutes les informations techniques de ponts et de routes est possible. Notre seconde série de tests concerne le calcul d'itinéraires où l'utilisateur choisit les lieux de départ et de destination, et le système fournit le chemin optimal. Le volet calcul d'itinéraire offre également la possibilité de fermer de tronçons des routes, puis il exclut ces derniers du calcul du trajet.

5.1 Démarrage et accueil

Avant d'accéder aux services proposés par le serveur et à la visualisation des résultats, il faut démarrer le serveur GeoServer. Ce démarrage ne nécessite pas un serveur Web comme Apache, car GeoServer joue le rôle de serveur Web en plus de son rôle de serveur cartographique. Il suffit donc de lancer GeoServer en sélectionnant l'exécutable `geoserver_start` à partir du répertoire '**bin**' de Geoserver. Une fois le démarrage terminé, on peut accéder aisément aux services par l'utilisation d'un navigateur Web. Pour ce faire, il suffit de saisir l'adresse du service sur la barre d'adresse du navigateur Web.

Une fois l'accès au système établi, l'outil affiche l'interface de la Figure 29. Cette interface présente cinq onglets dont le premier est l'interface de gestion des infrastructures routières et les quatre autres onglets sont utilisés pour le calcul d'itinéraires dont nous parlons plus tard. Elle présente également la légende, les fonctions d'agrandissement de l'affichage et une icône permettent d'afficher une liste de toutes les couches présentes dans l'application comme illustrée à droite sur la Figure 39. Ces couches sont les autoroutes, les routes locales, les ponts provenant des différentes sources et la carte de fond représentant l'environnement. Toutefois, il est possible d'enlever une, deux, trois ou toutes les couches de l'interface. Il est aussi possible de changer la carte de fond de l'interface en cochant la case de ladite couche.

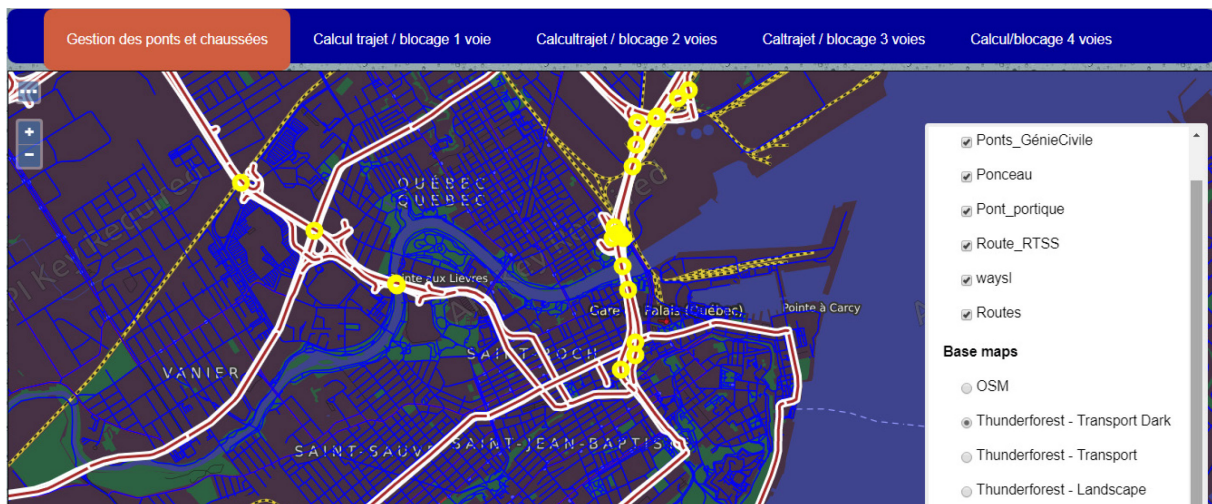


Figure 39 - Interface du système

5.2 Tests de consultation

Dans cette partie, quelques tests de consultation de données relatives aux infrastructures routières sont effectués. Ces tests concernent principalement la capacité du système d'afficher les informations de gestion des infrastructures suite à des recherches et des sélections. Ces tests ont pour but de montrer la fonctionnalité de cette partie de l'application. Il s'agit de naviguer sur la carte, choisir une des infrastructures représentées dans l'interface du système, et consulter les informations techniques de l'infrastructure. Cette infrastructure peut être une route, un pont ou un ponceau.

Test 1 : informations sur les routes

Les informations affichées aux Figures 40 et 41 sont le résultat obtenu après avoir sélectionné une route. Toutes les informations sur la route sont affichées. On utilise la barre de défilement pour découvrir les informations non visibles se trouvant plus bas. La Figure 40 montre la suite des informations une fois la barre défilée vers le bas.

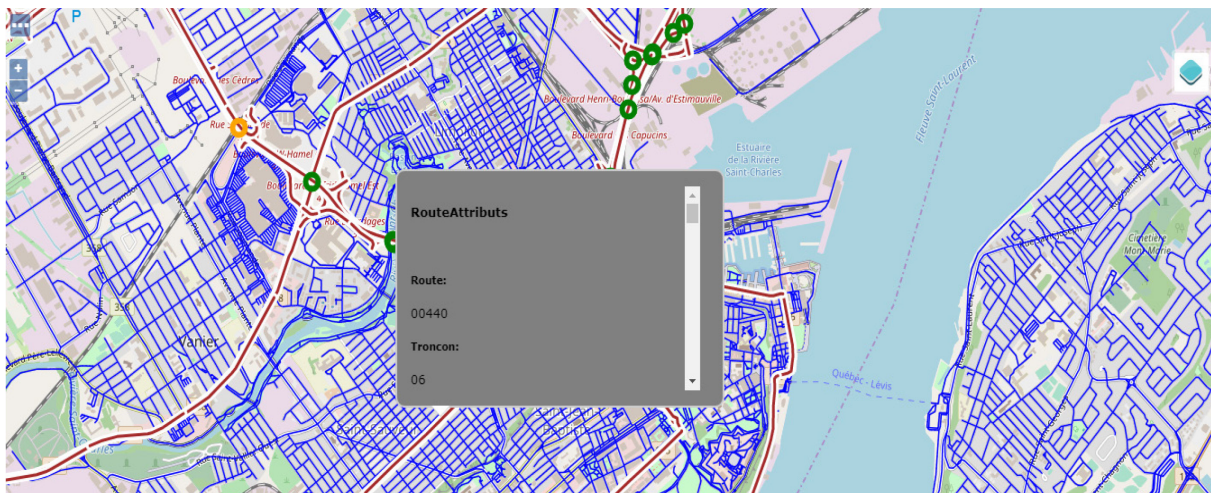


Figure 40 - Consultation d'un tronçon de route

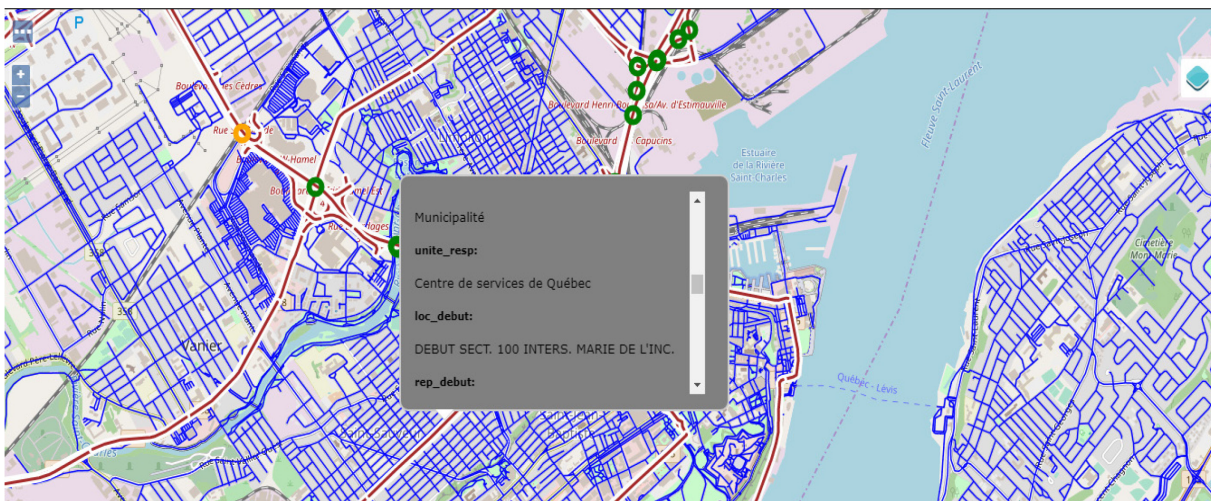


Figure 41 - Consultation d'un tronçon de route (suite)

Test 2 : informations sur les ponts

La Figure 42 montre la consultation d'un pont. Il s'agit d'un pont dont les données proviennent du CRGP. Ces ponts sont représentés par la couleur verte. Leurs données sont plus complètes que celles des ponts en orange qui représentent les ponts provenant du site Web du ministère des Transports du Québec (MTQ).

La Figure 43 présente des informations que l'on ne peut consulter que par défilement. Ainsi la Figure 43 représente les courbes de fragilité. Ces dernières déterminent l'état des ponts après un tremblement de terre. Elles permettent aux experts de déterminer si un pont est capable de supporter du trafic ou non.

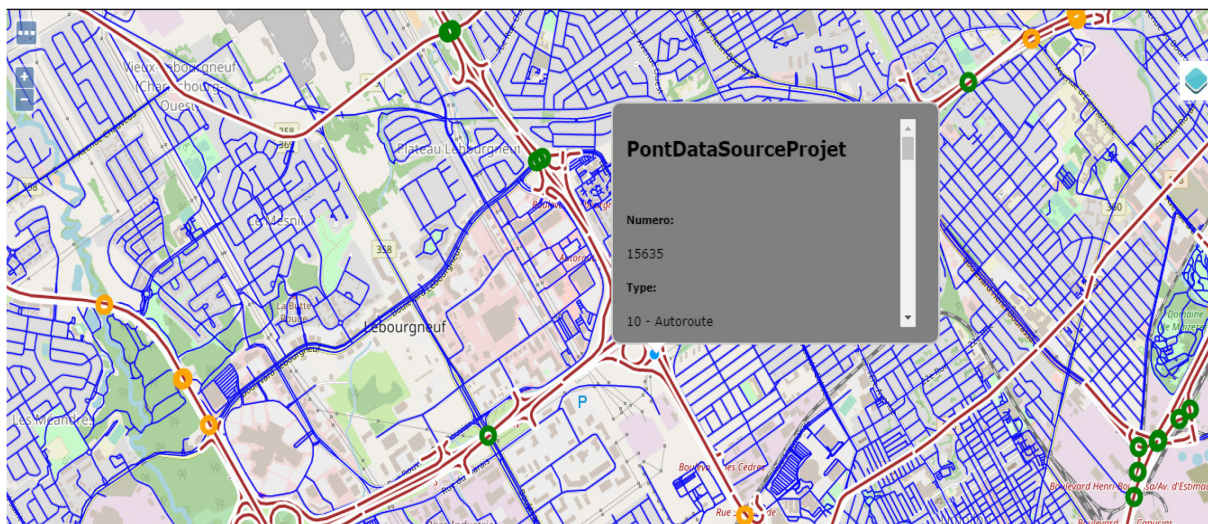


Figure 42 - Consultation de caractéristiques d'un pont vert

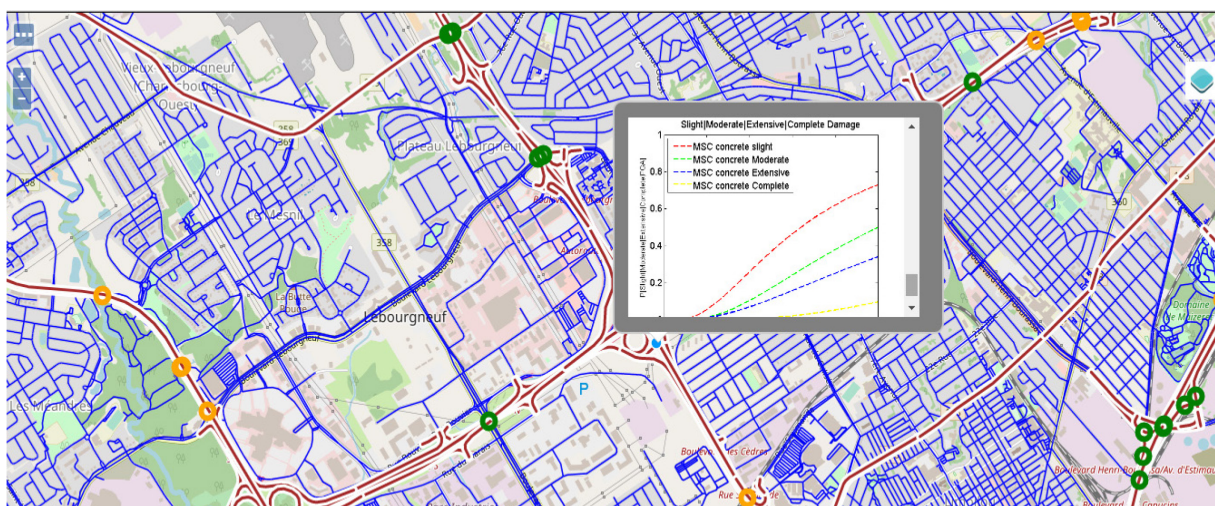


Figure 43 - Consultation de la courbe de fragilité d'un pont vert

Test 3 : informations sur les ponts complémentaires

À ce niveau, il s'agit des informations, qui portent sur des ponts, issus d'une source de données provenant du site Web du ministère des Transports du Québec (MTQ). Cette source ne dispose pas d'informations suffisantes décrivant tous les attributs d'un pont (Figures 44 et 45). Il faut préciser que ces ponts sont colorés en orange sur la carte. Encore une fois, un défilement permet de visualiser toutes les informations.

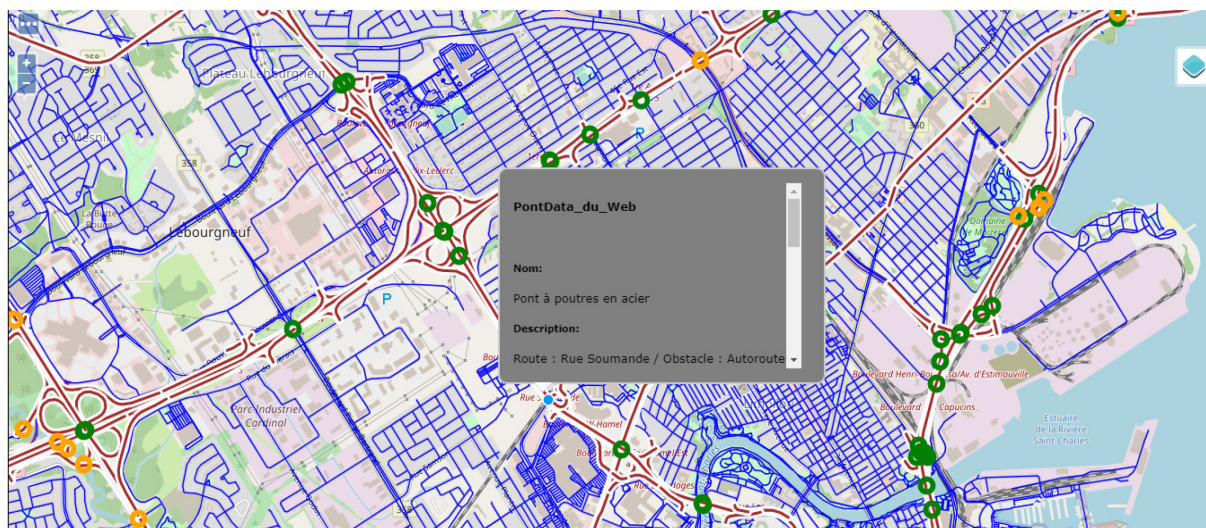


Figure 44 - Consultation de caractéristiques d'un pont marron



Figure 45 - Consultation de caractéristiques d'un pont marron (suite)

5.3 Tests de calcul d'itinéraires

Cette série de tests sert à montrer les capacités et le bon fonctionnement de l'outil servant au calcul d'itinéraires. Le calcul est accessible par l'intermédiaire de quatre onglets. Chaque onglet permet de tester le calcul selon une différente contrainte. Le premier onglet nous offre la possibilité de bloquer un tronçon de route, le deuxième onglet nous permet de bloquer deux tronçons de route, le troisième onglet nous permet de bloquer trois tronçons de route et le quatrième onglet nous permet de bloquer quatre tronçons de route.

Test 1 : blocage d'un tronçon

La Figure 46 montre un trajet où l'utilisateur a choisi un lieu de départ indiqué par le drapeau vert et un lieu d'arrivée représenté par le drapeau rouge. Le résultat du calcul d'itinéraire est illustré par le chemin en vert sur la carte.

La Figure 47 montre un itinéraire dans lequel il existe un chemin bloqué à cause d'un incident.

La Figure 48 présente une situation dans laquelle nous répétons le scénario précédent, mais en bloquant cette fois un tronçon parmi ceux de l'itinéraire qui vient d'être calculé.

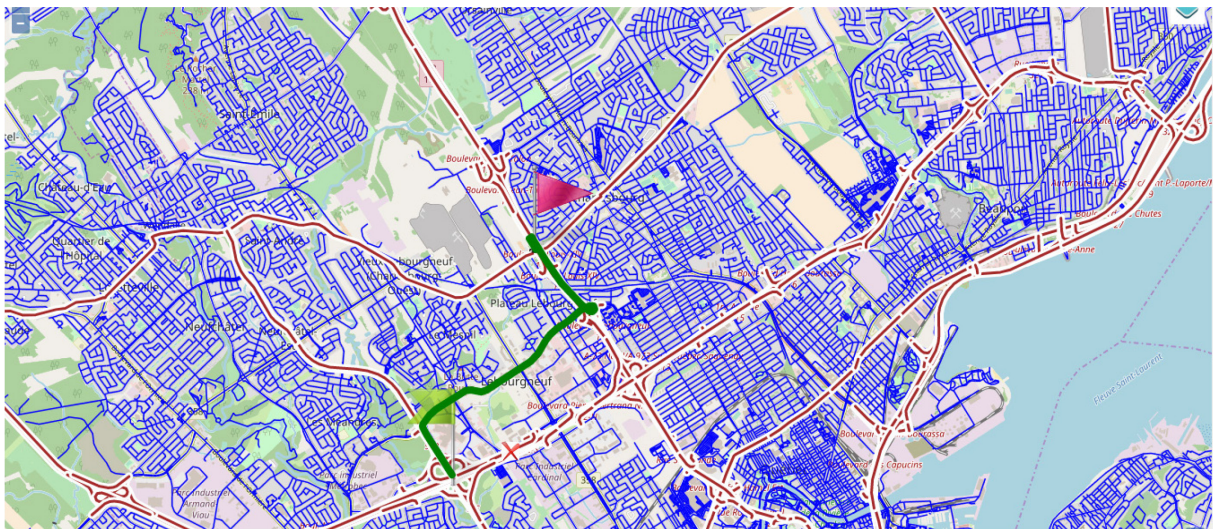


Figure 46 - Calcul d'itinéraire

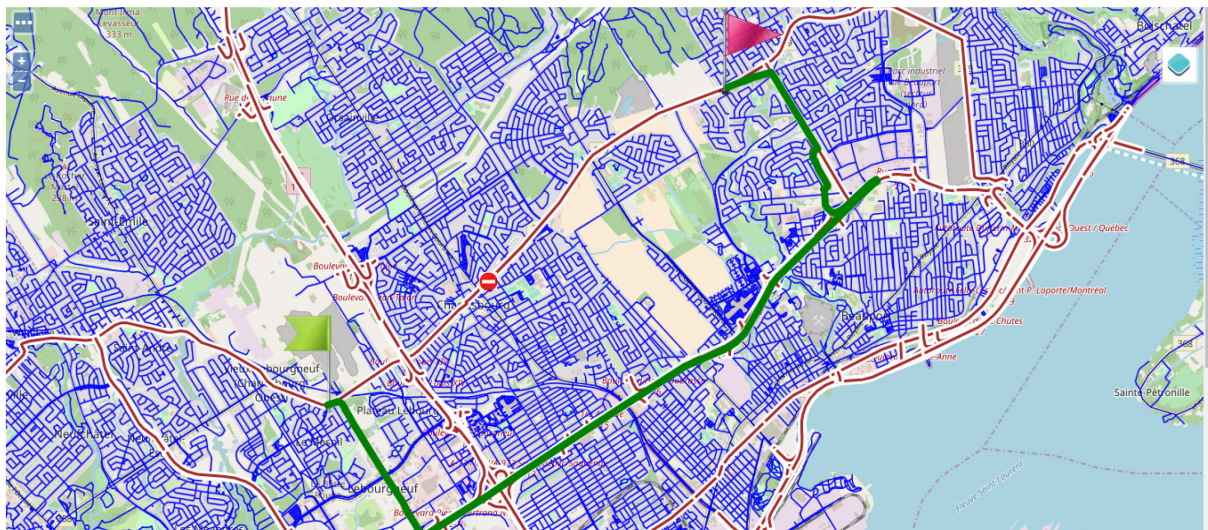


Figure 47 - Calcul d'itinéraire avec blocage d'un tronçon

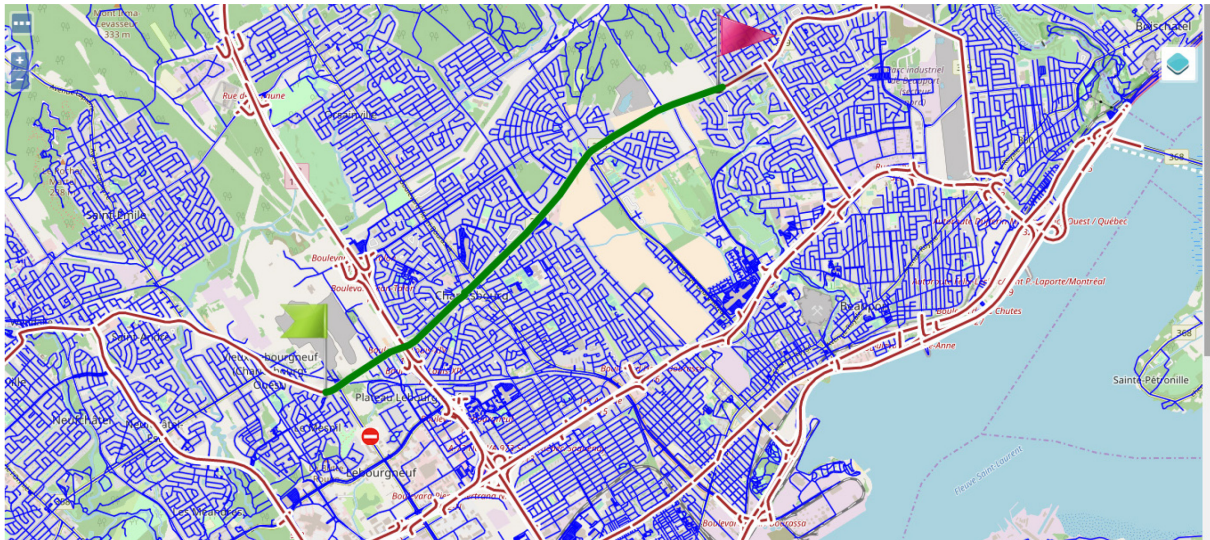


Figure 48 - Calcul d'itinéraire avec blocage d'un autre tronçon

Test 2 : blocage de deux tronçons

La Figure 49 montre le résultat d'un calcul d'itinéraire entre deux points suite au blocage de deux tronçons de route.

Le même trajet que celui de la Figure 49 est recalculé, mais cette fois des tronçons du trajet choisi sont bloqués pour montrer que l'algorithme va choisir effectivement d'autres tronçons de route pour ce nouveau trajet. Le résultat est présenté à la Figure 50.

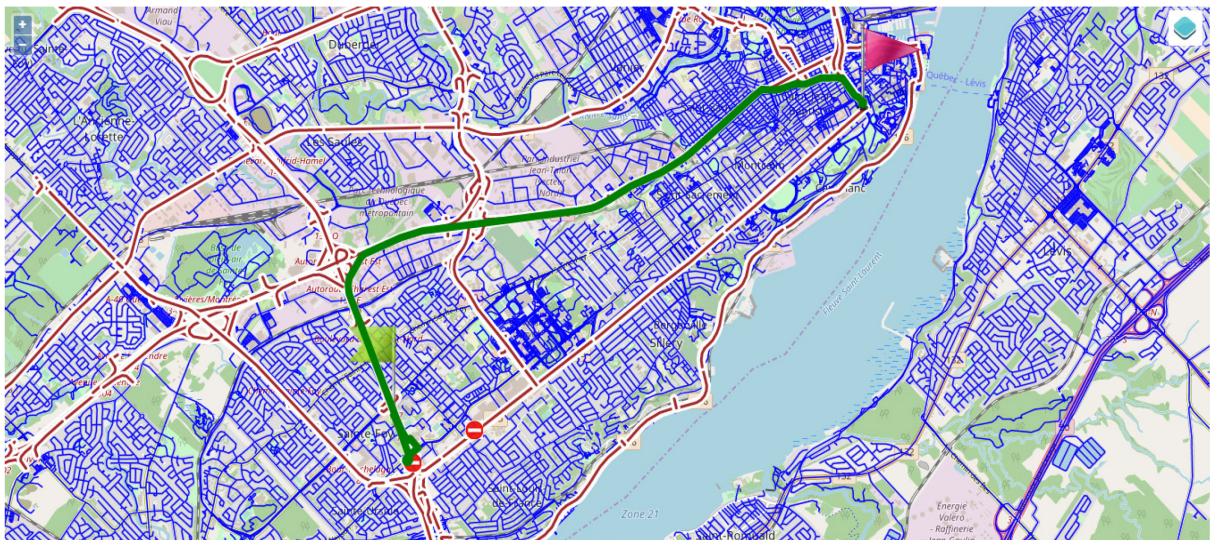


Figure 49 - Calcul d'itinéraire avec blocage de deux tronçons

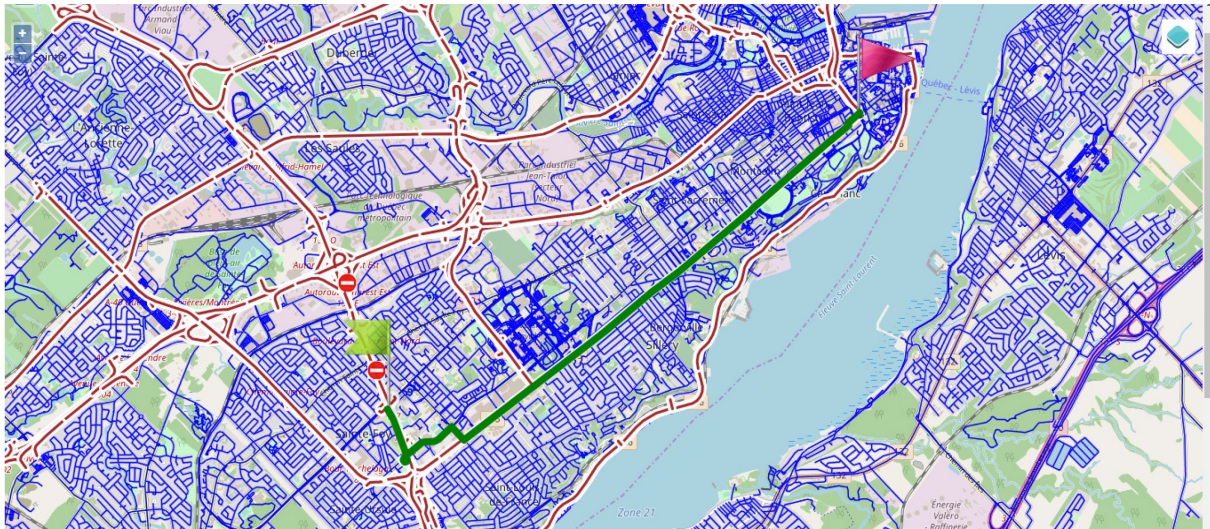


Figure 50 - Calcul d'itinéraire avec blocage de deux autres tronçons

Test 3 : blocage de trois tronçons

Pour ce test, nous procédons au calcul d'itinéraires en bloquant trois tronçons. La Figure 51 montre un scénario de calcul d'itinéraire sans blocage de tronçons de route.

La Figure 52 montre le résultat du calcul du même itinéraire suite aux blocages de trois tronçons. Elle montre bien qu'après le blocage de ces tronçons, le système est obligé de rechercher d'autres itinéraires.



Figure 51 - Calcul d'itinéraire avec blocage de trois tronçons



Figure 52 - Calcul d'itinéraire avec blocage de trois autres tronçons

Test 4 : blocage de quatre tronçons

Enfin, pour ce dernier test, nous testons le calcul d'itinéraire dans le cas où quatre tronçons sont bloqués. La figure 53 montre le calcul normal pour l'itinéraire choisi. La Figure 54 montre le nouveau calcul une fois quatre tronçons fermés.

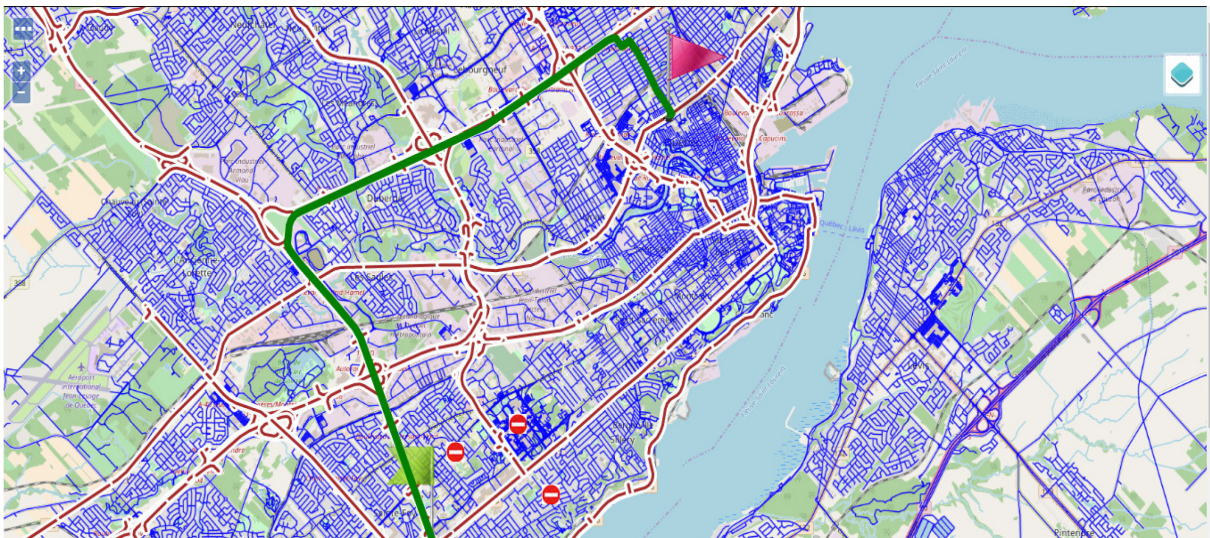


Figure 53 - Calcul d'itinéraire avec blocage de quatre tronçons

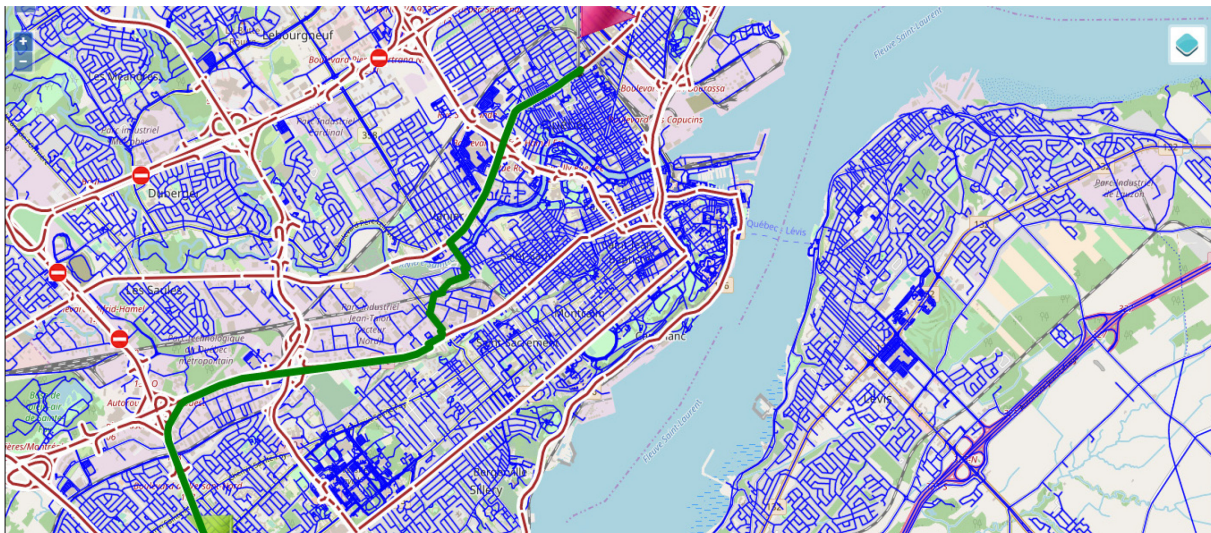


Figure 54 - Calcul d'itinéraire avec blocage de quatre autres tronçons

5.4 Analyse des résultats

Les résultats que nous avons obtenus répondent aux objectifs fixés. Cependant, nous avons rencontré plusieurs problèmes lors de la réalisation.

Le premier problème rencontré dans la phase de calcul d'itinéraire est l'absence des vitesses dans les données du réseau routier que nous avons téléchargé du site web du MTQ. Le MTQ protège ces données. Elles sont sous licence commerciale associée au produit « Adresses Québec ». Nous avons réussi à obtenir d'autres jeux de données ayant les vitesses sur le site de la ville de Québec, mais il s'est avéré que ces données ne sont pas conformes aux normes du MTQ. Par ailleurs, il n'est pas possible d'ajouter les limites des vitesses provenant des données de la ville de Québec aux données du MTQ, car la structure attributaire des données n'est pas la même. Nous n'avons pas trouvé un moyen de le faire, car il n'y a aucun attribut permettant de faire la projection entre les deux tables. Nous avons donc pris l'initiative d'ajouter les vitesses aux données, selon la classe de la route. Par exemple, si la classe de la route est « autoroute » alors une vitesse de 100km/h lui est attribuée en se basant sur le système d'indication des vitesses du MTQ.

Or, malgré l'ajout de vitesses, les données se sont avérées non adaptées au besoin, car elles n'ont pas été conçues pour faire des analyses telles que le calcul d'itinéraires. Ce problème peut être illustré de la manière suivante : pour créer une topologie (expliquée à la section « module de calcul d'itinéraire ») qui consiste à sectionner (séparer les tronçons de route aux intersections et ainsi pouvoir changer de route quand c'est nécessaire) les routes à chaque intersection. De plus, ces outils ne distinguent pas les intersections planes des intersections avec

étagement. Toutes les routes n'ont pas nécessairement besoin d'être sectionnées aux intersections avec étagement, car il existe des cas où on ne peut pas passer d'une route à l'autre aux intersections avec étagement. Il y a également des informations telles le sens unique, entre autres, qui manquent dans ces jeux de données. C'est ainsi que nous avons téléchargé les données d'Open Street Map (OSM) qui sont mieux adaptées pour ce genre d'analyse. De plus, elles sont compatibles à pgRouting, le test du calcul d'itinéraire avec ces données a réussi. Notons que les données d'Open Street Map (OSM) sont stockées dans une table de notre base de données, leur liaison avec les données de MTQ se fait à la base de coordonnées géographiques et elles sont utilisées uniquement pour faire le calcul d'itinéraire. Il faut également préciser que les données d'Open Street Map (OSM) ne disposent pas des informations qui sont nécessaires à la gestion de routes et de ponts. Nous devons donc combiner à la fois les données de MTQ et d'OSM.

Une fois l'algorithme testé et fonctionnel pour les données d'OSM, il ne reste qu'à implémenter le client Web. Pour tester la solution, on choisit d'abord les points d'origine et de destination via une sélection sur une carte électronique. Après la sélection, les coordonnées des deux points sont transmises à GeoServer. Ce dernier les transmet à pgRouting qui effectue le calcul de l'itinéraire et renvoie le résultat au client. Or quand les coordonnées sont transmises à GeoServer après la sélection des points d'origine et de destination, le résultat ne s'affiche pas sur la carte. Après avoir vérifié le système de coordonnées, vérifié de la bonne transmission des coordonnées des points et changé le mode transmission des coordonnées des points à GeoServer, nous avons pu finalement détecter que le problème était lié aux coordonnées de points. L'expression régulière, présente dans les paramètres de la vue, qui est chargée de contrôler (s'assurer de la conformité de la chaîne de caractères attendue) les coordonnées des points envoyées par l'utilisateur à GeoServer était incorrecte. Nous avons résolu le problème en réécrivant correctement l'expression régulière.

Enfin, le dernier problème concerne l'élimination des tronçons de route via des sélections lors d'un incident (effondrement d'un pont ou travaux sur ce dernier). Le code semble être bien écrit, car, lors de l'exécution aucune erreur n'est signalée. Par contre le résultat produit est incorrect. Après analyse, nous avons trouvé que pgRouting utilisait la mauvaise table (table de sommets) pour calculer l'itinéraire au lieu d'utiliser la table de tronçons. Nous avons résolu ce problème en forçant PgRouting à utiliser la table de tronçons.

Chapitre 6 Conclusion

Rappelons-nous que nos objectifs sont de gérer les infrastructures en rendant les informations les concernant accessibles et d'effectuer des calculs d'itinéraires en cas d'incidents sur les routes. Selon les tests que nous avons effectués et les résultats satisfaisants obtenus, nous pensons que nos objectifs fixés sont atteints. Ces résultats obtenus sont tels que les experts de génie civil peuvent consulter de manière très simple les informations relatives aux routes et ponts, afin de planifier leurs actions. Il faut ajouter que les courbes de fragilité associées à chaque type des ponts y sont intégrées pour faciliter la prévision des états des ponts après un séisme. Finalement, quand un pont s'effondre, ou qu'il est déclaré à risque après son évaluation via sa courbe de fragilité, ou bien même qu'il y ait juste des travaux planifiés, le système permet de fermer les tronçons concernés, et de calculer l'itinéraire en excluant ces tronçons pour fournir un trajet optimal. Au vu de ce qui précède, on peut dire que les objectifs fixés sont atteints.

De plus, le système est accessible à distance par le biais d'un simple navigateur Web. Nous croyons qu'il est très simple d'utilisation, puisque toutes les manipulations se font par de simples sélections sur une carte représentant les objets. Cette interface graphique est conviviale et intuitive, car elle représente les entités manipulées tout comme leur environnement au voisinage, de façon très proche de la réalité grâce à l'usage du SIG.

Ainsi, on peut dire que le cheminement suivi pour le développement du système est satisfaisant. Il faut également ajouter que les outils utilisés dans le cadre de la réalisation de ce travail sont libres d'accès.

Toutefois, il faut reconnaître que le travail réalisé nécessite quelques améliorations. C'est pourquoi les suggestions suivantes sont formulées en guise de perspectives.

Il faudrait ajouter des fonctions qui permettraient d'intégrer des modèles de prévision tels que ceux cités dans la revue de littérature de ce mémoire (PQI, D et T_n). Ces modèles devraient fournir l'état de chaque entité de façon simple et claire afin d'aider les décideurs. Ainsi, il est possible de présenter les ponts par des différentes couleurs selon l'état de chaque pont. C'est-à-dire mettre en exergue leurs états à travers les couleurs. Par exemple, si un pont est de couleur rouge, cela signifierait que son état est critique, donc son utilisation n'est pas recommandée ; il devrait être dangereux. Par conséquent, il faut envoyer des agents sur place pour voir les solutions appropriées à apporter. Au même moment, ces modèles pourraient bien être combinés à d'autres, comme les modèles fournissant les courbes de fragilités déjà présentes dans ce travail.

Il faudrait également intégrer dans l'avenir les courbes de fragilité dans le processus du calcul d'itinéraires. C'est-à-dire, si dans un trajet il existe un tronçon comportant un pont ne pouvant pas supporter le trafic après un tremblement de terre, selon sa courbe de fragilité, ce qui signifierait que le pont est susceptible à l'effondrement, il faudrait que ce tronçon comprenant ce pont soit exclu du processus de calcul de l'itinéraire.

Nous recommandons également d'enrichir l'équation de calcul d'itinéraire en intégrant des éléments tels que la congestion, les zones d'école, etc.

Enfin, notre système est développé pour la ville de Québec, il faudrait donc l'étendre pour couvrir à toute la province.

Annexe A

Présentation des données

A.1 Tableau 9 - Réseau routier (Source de données, Cartothèque de l'UDS et ministère des Transports, Division de la géomatique, 2015)

Objet	Projection	Attributs	Description
Route1 (560 enregistrements)	WGS 84 (EPSG: 4326)	Route	Numéro de la route
		Tronçon	Numéro du tronçon
		Section	Numéro de la section
		Code	Code de sous-route
		Séquence	Séquence de la sous-route
		Sous_code	Sous-code de la sous-route
		Cote_chaus	Côté de la chaussée (C, D, G
		Longueur	Longueur (m) de la section de la route
		source_mes	Source de la mesure : CDAO, GPS, LIDAR, Longueur estimée, Odomètre
		Statut	Statut de la sous-route : A (Actif), F (Fermée), R (En réfection), T (Transitoire)
		cl_fonct	Classe fonctionnelle : Autoroute, Nationale, Régionale, Collectrice
		Nom	Nom de la sous-route
		auto_resp	Autorité responsable : MTQ (Ministère des Transports du Québec) Municipal (Municipalité)

			<p>PJCCI (Les Ponts Jacques-Cartier Champlain Inc.)</p> <p>A30 (PPP: Nouvelle Autoroute 30 S.E.N.C.)</p> <p>CA25 (PPP: Concession A25 S.E.C.)</p> <p>MERN (Ministère de l'Énergie et des Ressources naturelles)</p> <p>TC (Transport Canada)</p> <p>CN (Canadien National)</p> <p>MRC (Municipalité régionale de comté)</p> <p>PP Routier (Partenaire parc routier)</p> <p>ASFC (Agence des services frontaliers du Canada)</p> <p>SAAQ (Société de l'Assurance automobile du Québec)</p> <p>Municipal</p> <p>etc.</p>
		type_autor	Type d'autorité : Gouvernement du Québec, Gouvernement fédéral, PPP, Inconnu, Municipalité
		DG	Numéro de la direction générale (DG) responsable de la sous-route
		DT	Numéro de la direction territoriale (DT) responsable de la sous-route

	CS	Numéro du centre de services responsable de la sous-route
	SC	Numéro du sous-centre de services responsable de la sous-route
	unite_resp	Nom de l'unité responsable de la sous-route
	loc_debut	Localisation du début de la sous-route
	rep_debut	Type de repère pour délimiter le début de la sous-route
	pos_debut	Décalage du repère (m) par rapport au début de la sous-route
	loc_fin	Localisation de la fin de la sous-route
	rep_fin	Type de repère pour délimiter la fin de la sous-route
	pos_fin	Décalage du repère (m) par rapport à la fin de la sous-route
	odo_orig	Odonyme d'origine
	odo_dest	Odonyme de la destination
	Debut	Date de la période d'opération de la sous-route
	rte_chev	Chevauchement de routes
	Cat_Reseau	Catégorie (s) de réseau liée (s) à la sous-route
	dern_modif	Dernière modification effectuée sur la sous-route

A.2 Tableau 10 - Réseau routier (Source de données, OSM)

Objet	Projection	Attributs	Description
Route2 (123883 enregistrements)	WGS 84 (EPSG: 4326)	Gid	Identifiant de l'enregistrement
		Class_id	
		Length	Longueur de segment
		Name	Nom de segment
		Source	Début de segment
		Target	Fin de segment
		X1 Y1 X2 Y2	Cordonnées de début et fin de segment
		Cost	Coût de l'allée
		Reverse_cost	Coût de retour
		One_way	Pour savoir si sens unique
		Maxspeed_forward	Vitesse maximal allée
		Maxspeed_backward	Vitesse maximal retour
		Osm_id	Identifiant de l'objet osm
		Priority	Priorité du segment
		The_geom	Géométrie du segment

A.3 Tableau 11 - Ponts (Source de données, initiateurs du projet)

Objet	Projection	Attributs	Description
Pont (127 enregistrements)	WGS 84 (EPSG: 4326)	NO STRUCTURE	Identifiant de l'enregistrement
		CLASSE DE ROUTE	Classe De Route
		TYPE STRUCTURE	Type Structure

	Nom	Nom du pont
	ROUTE	Route portée par le pont
	OBSTACLE	L'obstacle traversé
	MUNICIPALITE	La municipalité où est situé le pont
	NBRE TRAVÉE	Nombre de travée du pont
	TYPE TRAVEE	Type de travée
	LATITUDE LONGITUDE	Coordonnées de localisation du pont
	AN CONST UNITE FONDATION	Année de construction de la fondation du pont
	AN CONST SYSTEME STRUCTURAL	Année de construction de la structure du pont
	AN CONST PLATELAGE	Année de construction du platelage
	LONG TABLIER (mm)	Longueur du tablier du pont
	LARG HORS TOUT (mm)	Largeur hors tout
	ANGLE BIAIS	L'angle biais

A.4 Tableau 12 - Ponts (Source de données, Site web du Ministère des Transports, Division de la géomatique, 2015)

Objet	Projection	Attributs	Description
Pont2 (55 enregistrements)	WGS 84 (EPSG: 4326)	Gid	Identifiant de l'enregistrement
		Longitude	Coordonnées de localisation du pont
		Latitude	

		Name	Nom du pont
		Descriptio	Renseigne la route portée, l'obstacle surpassé, la charge à supporter et l'année de construction
		Geom	Géométrie du pont

Bibliographie

- [1] Transport Québec, “Gestion des données géospatiales au MTQ,” CRG / Réseau Convergence, Mars 2015
- [2] R. Kakade, “Road network analysis using geoinformatic technique for Akola city, Maharashtra state, India,” *International Journal of Engineering Research & Technology*, vol. 2, pp. 1884–1886, 2013
- [3] C. Dunn and D. Newton, “Optimal routes in GIS and emergency planning applications,” *Area*, vol. 24, no. 3, pp. 259–267, 1992.
- [4] Alazab, Ammar, Venkatraman, Sitalakshmi, Abawajy, Jemal and Alazab, Mamoun, “An Optimal Transportation Routing Approach using GIS-based Dynamic Traffic Flows,” *International Conference on Management Technology and Applications*, Singapore, pp. 172-178, 2010.
- [5] N. M. Waters, “Transportation GIS: GIS-T,” *Geographical information systems: Principles, techniques, management and applications*, pp. 827-844, 1993.
- [6] S. Zhu, D. Levinson, H. Liu, K. Harder, and A. Danczyk, “Traffic Flow and Road User Impacts of the Collapse of the I-35W Bridge over the Mississippi River,” *Minnesota Department of Transportation Research*, 2009.
- [7] Conseil des ministres responsables des transports et de la sécurité routière. "Le coût élevé de la congestion dans les villes canadiennes", consulté en Décembre 2017. <http://www.comt.ca/french/uttfcongestion-2012f.pdf>
- [8] Y.Itoh, Y.Shintoku, A.Hammad, Liu Chunlu, “Network-Level Bridge Life-Cycle Management System,” *Journal of Infrastructure Systems*, Vol. 3, pp. 31–39, 1997.
- [9] H. a. Karimi, L. Zhang, and J. G. Benner, “Personalized accessibility map (PAM): a novel assisted wayfinding approach for people with disabilities,” *Annals of GIS*, vol. 20, pp. 99–108, 2014.
- [10] H. Bast, D. Delling, A. V Goldberg, T. Pajor, M. Müller-hannemann, D. Wagner, and R. F. Werneck, “Route Planning in Transportation Networks (Karlsruhe Institute of Technology),” *Microsoft Research Technical Report*, arXiv:1504.05140v1, pp. 1–65, 2014.

- [11] D. Schultes, "Route Planning in Road Networks," *Ausgezeichnete Inform.*, pp. 1–235, 2008.
- [12] M. Abousaeidi, R. Fauzi, and R. Muhamad, "Geographic Information System (GIS) modeling approach to determine the fastest delivery routes," *Saudi Journal of Biological Sciences*, vol. 23, pp. 555–64, 2015.
- [13] A. Osvald and L. Z. Stirn, "A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food," *Journal of Food Engineering*, vol. 85, pp. 285–295, 2008.
- [14] C. D. Tarantilis and C. T. Kiranoudis, "Distribution of fresh meat," *Journal of Food Engineering*, vol. 51, pp. 85–91, 2002.
- [15] C. D. Tarantilis and C. T. Kiranoudis, "A meta-heuristic algorithm for the efficient distribution of perishable foods," *Journal of Food Engineering*, vol. 50, pp. 1–9, 2001.
- [17] L. C. Coelho, J. Renaud, and G. Laporte, "Road-based goods transportation: a survey of real-world logistics applications from 2000 to 2015," *INFOR: Information Systems and Operational Research*, vol. 34, pp. 1–18, 2016.
- [18] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [19] R. Gutman, "Reach-based routing: A new approach to shortest path algorithms optimized for road networks," *Proceedings of the Sixth on Algorithm EngineerWorkshoping and Experiments and the First Workshop on Analytic Algorithms and Combinatorics*, pp. 100–111, 2004.
- [20] Quora "How does Google Maps calculate your ETA?", consulté en juin 2017. <https://www.quora.com/Speed-Limits/How-does-Google-Maps-calculate-your-ETA>
- [21] Uber Engineers "How Uber Engineers an Efficient Route", consulté en juin 2017. <https://eng.uber.com/engineering-an-efficient-route/>
- [22] Tomtom, "About IQ ROUTES", consulté en juin 2017. [http://uk.support.tomtom.com/app/answers/detail/a_id/9552/~about-iq-routes/](http://uk.support.tomtom.com/app/answers/detail/a_id/9552/~/about-iq-routes/)

- [23] Waze, “Routing server”, consulté en juin 2017. https://wiki.waze.com/wiki/Routing_server
- [24] M. Baum, J. Dibbelt, T. Pajor, and D. Wagner, “Energy-optimal routes for electric vehicles,” Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL'13, vol. 2013, pp. 54–63, 2013.
- [25] H. Bast, S. Funke, D. Matijevic, P. Sanders, and D. Schultes, “In Transit to Constant Time Shortest-Path Queries in Road Networks,” Proc. of the Workshop on Algorithm Engineering and Experiments, pp. 45–59, 2007.
- [26] G. V. Batz and P. Sanders, “Time-dependent route planning with generalized objective functions,” Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, vol. 7501, pp. 169–180, 2012.
- [27] D. Delling, T. Pajor, and D. Wagner, “Accelerating multi-modal route planning by access-nodes,” Department of Computer Science, Universität Karlsruhe, vol. 5757, pp. 587–598, 2009.
- [28] M. Hilger, E. Kohler, R. H. Mohring, and H. Schilling, “Fast Point-to-Point Shortest Path Computations with Arc-Flags,” Institute of Mathematics, TU Berlin, Germany, vol. 74, pp. 41–72, 2009.
- [29] E. Horvitz and J. Krumm, “Some Help On the Way : Opportunistic Routing Under Uncertainty,” Proc. 2012 ACM Conf. Ubiquitous Comput. - UbiComp '12, pp. 371–380, 2012.
- [30] Coordonnées gps, “Calcul d'itinéraire Google Maps”, consulté en juin 2017. <https://www.coordonnees-gps.fr/itineraire>
- [31] Itilog, “Calcul d'itinéraire Google Maps”, consulté en juin 2017. <http://www.itilog.com/>
- [32] Plans, “Une meilleure façon d’arriver à destination”, consulté en juillet 2018. <https://www.apple.com/ca/fr/ios/maps/>
- [33] Wikipédia, “Apple Plans”, consulté en juin 2018.

https://fr.wikipedia.org/wiki/Apple_Plans

[34] Le moniteur automobile, “Quel site de cartographie pour préparer son itinéraire”, consulté en juillet 2018. <https://www.moniteurautomobile.be/actu-auto/mobilite/cartographie-le-meilleur-viamichelin-mappy-google-maps-waze-app-routeplanner.html>

[35] CNET, “Apple Plans, Google Maps, Waze : quelle application pour trouver sa route”, consulté en juillet 2018. <http://www.cnetfrance.fr/produits/apple-plans-google-maps-waze-quele-application-pour-trouver-sa-route-39802267.htm>

[36] Geoffroy Ondet “Les meilleures applications pour la route”, consulté en juillet 2018. <http://www.01net.com/actualites/les-meilleures-applications-pour-la-route-627888.html>

[37] Sygic, “La navigation la plus évoluée”, consulté en juillet 2018. <http://www.sygic.com/fr/gps-navigation>

[38] Wikipédia “Bing Maps”, consulté en juillet 2018. https://en.wikipedia.org/wiki/Bing_Maps

[39] Wikipédia “TomTom”, consulté en juillet 2018. <https://fr.wikipedia.org/wiki/TomTom>

[40] Wikipédia “Garmin”, consulté en juillet 2018. <https://fr.wikipedia.org/wiki/Garmin>

[41] A. C. Lemer, “Progress Toward Integrated Infrastructure-Assets-Management Systems: GIS and Beyond,” APWA International Public Works Congress, pp. 7–24, 1998.

[42] Q. Li and F. Xie, “Expressway Maintenance Decision Support System Based on GIS,” WASE International Conference on Information Engineering, pp. 139–141, 2010.

[43] F. Sandrone and V. Labiouse, “Identification and analysis of Swiss National Road tunnels pathologies,” Tunnelling Underground Space Technology, vol. 26, pp. 374–390, 2011.

[44] A. Ibraheem and D. Falih, "Applying Geographic Information System (GIS) for Maintenance Strategy Selection,"Engineering, Vol. 4, 2012, pp. 44-54.

[45] S.-F. Jiang and C. Zhang, “Design and Implementation of a WEBGIS-Based Quality

Evaluation System for Bridge Construction,” Information Science and Engineering (ICISE), 1st International Conference on, pp. 4257–4261, 2009.

[46] X. Zhang, G. Li, and X. Lan, “Research on WebGIS Performance Optimization,” 7th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–4, 2011.

[47] Q.-S. LI. and D.-P. HE, “Application of GIS in Highway Maintenance Management,” The Twelfth COTA International Conference of Transportation Professionals, pp. 2433–2442, 2012.

[48] M. Johanns and J. Craig, “Pavement Maintenance Manual,” Departement of road, Nebraska, 2002.

[49] M. Ge, L. Sun, Y. Pan, and Y. Li, “Design and Implementation of Freeway Infrastructure Safety and Emergency Management System,” Procedia – Social and Behavioral Sciences, vol. 96, pp. 728–740, 2013.

[50] M. R. Halfawy, L. A. Newton, and D. J. Vanier, “Review of Commercial Municipal Infrastructure Asset Management Systems,” Electronic Journal of Information Technology in Construction, vol. 11, pp. 211–224, 2006.)

[51] Y.Jiao, Z .Bowen and H.Siranc, “Study on Application of a Management System for Pavement Based on iPad Terminal with GIS“, Information Technology Journal, pp. 520-523, 2012

[52] Z. Y. Hong and A. Dong, “Research and Development of WebGIS Based on MapXtreme\slash Java,” Shenyang Institute of Technology, vol. 22, no. 3, pp. 45–48, 2003.

[53] M. Wang and P. Yue, “Research and Realization of Highway Project Management Information System Based on GIS,” 2008 4th 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 402–406, 2008.

[54] X. Yu, J. Song, F. Yang, Y. Dong, and Z. Dong, “Study of Intelligent Public Transport Monitoring System Based on WebGIS,” 2009 Second International Conference on Intelligent Networks and Intelligent Systems. pp. 78–81, 2009.

[55] Z. Jia, X. Han “Research and Implementation of GIS Based on Web,” International

Conference on Computer Application and System Modeling (ICCASM), pp. 526–529, 2010.

[56] S. Rao and S. Vinay “Choosing the right GIS framework for an informed Enterprise Web GIS Solution,” 2009.

[57] S. Agrawal and R. D. Gupta, “Development and Comparison of Open Source based Web GIS Frameworks on WAMP and Apache Tomcat Web Servers,” ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. XL, no. 4, pp. 1–5, 2014.

[58] S. Kubota, K. Matsumura, S. Yano, T. Kitadani, I. Kitagawa and A. Ichiuji, “Disaster Information Sharing System Using Open Source Web GIS,” Computing in Civil and Building Engineering, pp. 1254–1261, 2014.

[59] L. Ford et D. Fulkerson “Flows in Networks,” Princeton University Press, 1962.

[60] M. Gendreau, G. Laporte, and R. Séguin, "An exact algorithm for the vehicle routing problem with stochastic customers and demands", *Transportation Science* 29, 143- 155, 1996.

[61] D. Roberts and E. Hadjiconstantinou, "The vehicle routing problem with stochastic demands: An overview. Working paper", Imperial College Management School, London, 1996.

[62] A. S. Kenyon and D. P. Morton, "Stochastic Vehicle Routing with Random Travel Times", *Transportation Science* vol. 37, pp. 69–82, 2003.

[63] P. Nicoară, and I. Haidu, "A GIS based network analysis for the identification of shortest route access to emergency medical facilities", *Geographia Technica* vol. 09, pp. 60–67.

[64] Service Technique de l'Urbanisme : "Systèmes d'Information géographique : des concepts aux réalisations". Edition Hermès, 1991.

[65] Becker R. et al. "Network Visualization". 4th International Symposium on Spatial Data Handling, Zürich, Switzerland, 23-27 Juillet 1990.

[66] Ian L. McHarg, "Design with Nature ", Garden City, NY, Natural History Press. 1969.

[67] Antonio Santiago, The book of OpenLayers 3 Theory & Practice, 2015.
file:///C:/Users/muhammed/Downloads/thebookofopenlayers3-sample%20(1).pdf

[68] Béné, G.B., "Système d'information géographique", 2002, Note de cours, Université de Sherbrooke, Sherbrooke.

[69] Ramez Elmasri et Shamkant Navathe, "Conception et architecture des bases de données" Pearson, 4e édition ISBN 978-2-7440-7248-2

[69] PostGIS, "Spatial and Geographic objects for PostgreSQL", consulté en novembre 2017. <http://postgis.net/>

[70] André Légaré & ASSOCIÉS INC., "L'entretien du réseau routier du Québec, l'exigence de solutions durables," 2005, consulté en septembre 2018.
http://www.andrelegare.com/pdf_FR/Coalition-etude.pdf

[71] Transports Québec, "Classification fonctionnelle", consulté en septembre 2018.
http://www.bape.gouv.qc.ca/sections/mandats/berges_richelieu/documents/DA3.1.pdf

[72] Transports Québec, "Guide de la codification et du mesurage du réseau routier", consulté en septembre 2018. <http://www.bv.transports.gouv.qc.ca/mono/0993494.pdf>

[73] QGIS, "Système d'Information Géographique Libre et Open Source", consulté en septembre 2018. <https://qgis.org/fr/site/>

[74] ArcMap, "Prise en main d'ArcMap", consulté en septembre 2018.
<http://desktop.arcgis.com/fr/arcmap/10.5/get-started/main/get-started-with-arcmap.htm>

[75] M. Głobowski et al., "Efficiency Evaluation of Shortest Path Algorithms," IARIA, pp. 154–160, 2013.

[76] C. Prins, "Comparaison d'algorithmes de plus courts chemins sur des graphes routiers de grande taille (*)," *païro*, tome 30, pp. 333–357, 1996.